

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-98-

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-018).

0807

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December, 1995	3. REPORT TYPE Final
4. TITLE AND SUBTITLE USAF Summer Research Program - 1995 High School Apprenticeship Program Final Reports, Volume 15B, Wright Laboratory		5. FUNDING NUMBERS
6. AUTHORS Gary Moore		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Research and Development Labs, Culver City, CA		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NI 4040 Fairfax Dr, Suite 500 Arlington, VA 22203-1613		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES Contract Number: F49620-93-C-0063		
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release		12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) The United States Air Force High School Apprenticeship Program's (USAF- HSAP) purpose is to place outstanding high school students whose interests are in the areas of mathematics, engineering, and science to work in a laboratory environment. The students selected to participate in the program work in an Air Force Laboratory for a duration of 8 weeks during their summer vacation.		
Reproduced From Best Available Copy		
14. SUBJECT TERMS AIR FORCE HIGH SCHOOL APPRENTICESHIP PROGRAM, APPRENTICESHIP, AIR FORCE RESEARCH, AIR FORCE, ENGINEERING, LABORATORIES, REPORTS, SCHOOL, STUDENT, SUMMER, UNIVERSITIES		15. NUMBER OF PAGES
		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified
20. LIMITATION OF ABSTRACT UL		

THIS QUANTITY REPRESENTS 3

UNITED STATES AIR FORCE
SUMMER RESEARCH PROGRAM -- 1995
HIGH SCHOOL APPRENTICESHIP PROGRAM FINAL REPORTS

VOLUME 15B
WRIGHT LABORATORY

RESEARCH & DEVELOPMENT LABORATORIES
5800 Uplander Way
Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Major David Hart

Program Manager, RDL
Scott Licoscas

Program Administrator, RDL
Gwendolyn Smith

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Bolling Air Force Base
Washington, D.C.
December 1995

19981231 106

PREFACE

Reports in this volume are numbered consecutively beginning with number 1. Each report is paginated with the report number followed by consecutive page numbers, e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

Due to its length, Volume 15 is bound in two parts, 15A and 15B. Volume 15A contains #1-26. Volume 15B contains reports #28-52. The Table of Contents for Volume 15 is included in both parts.

This document is one of a set of 16 volumes describing the 1995 AFOSR Summer Research Program. The following volumes comprise the set:

<u>VOLUME</u>	<u>TITLE</u>
1	Program Management Report
	<i>Summer Faculty Research Program (SFRP) Reports</i>
2A & 2B	Armstrong Laboratory
3A & 3B	Phillips Laboratory
4	Rome Laboratory
5A, 5B, & 5C	Wright Laboratory
6A & 6B	Arnold Engineering Development Center, Wilford Hall Medical Center and Air Logistics Centers
	<i>Graduate Student Research Program (GSRP) Reports</i>
7A & 7B	Armstrong Laboratory
8	Phillips Laboratory
9	Rome Laboratory
10A & 10B	Wright Laboratory
11	Arnold Engineering Development Center, Wilford Hall Medical Center and Air Logistics Centers
	<i>High School Apprenticeship Program (HSAP) Reports</i>
12A & 12B	Armstrong Laboratory
13	Phillips Laboratory
14	Rome Laboratory
15A&15B	Wright Laboratory
16	Arnold Engineering Development Center

HSAP FINAL REPORT TABLE OF CONTENTS

i-xiv

1. INTRODUCTION	1
2. PARTICIPATION IN THE SUMMER RESEARCH PROGRAM	2
3. RECRUITING AND SELECTION	3
4. SITE VISITS	4
5. HBCU/MI PARTICIPATION	4
6. SRP FUNDING SOURCES	5
7. COMPENSATION FOR PARTICIPATIONS	5
8. CONTENTS OF THE 1995 REPORT	6

APPENDICIES:

A. PROGRAM STATISTICAL SUMMARY	A-1
B. SRP EVALUATION RESPONSES	B-1

HSAP FINAL REPORTS

**Applications of Electronic Devices for use in
Electronic Warfare and Digital Signal Processing**

Jason M. Leopold

Senior

**Bellbrook High School
3491 Upper Bellbrook Rd.
Bellbrook, Ohio 45305**

**Sponsored by: Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.**

and

Wright Laboratories

August 1995

Applications of Electronic Devices for use in
Electronic Warfare and Digital Signal Processing

Jason M. Leopold
Senior
Bellbrook High School

Abstract

Various electronic devices were studied to learn of their use in electronic warfare and digital signal processing. Mathcad, DADiSP, Matlab, GW Basic, and HT Basic were some of the software tools used. Circuit building, program writing, and IEEE-488 interfacing were also studied. One of the programs written is included. Two side projects are also discussed: the interface with an ADI-DSP card, and the linking of two computers so that all of the drives on both computers can be used on one.

Applications of Electronic Devices for use in
Electronic Warfare and Digital Signal Processing

Jason M. Leopold

Many different electronic devices and software tools are now being used by engineers. These necessary tools drastically reduce the time spent on projects which require many calculations. Many of these tools are software tools which require the use of a computer. Therefore, an engineer must be fluent in many different computer languages, as well as patient to learn the ones he or she does not know. The computer is a necessary tool for many reasons. Not only can the computer be used to make many complicated calculations, but it can also be used to control the other equipment, such as spectrum analyzers, oscilloscopes, frequency generators, etc. through the use of an IEEE-488 interface. This way, all work can be done without leaving your chair.

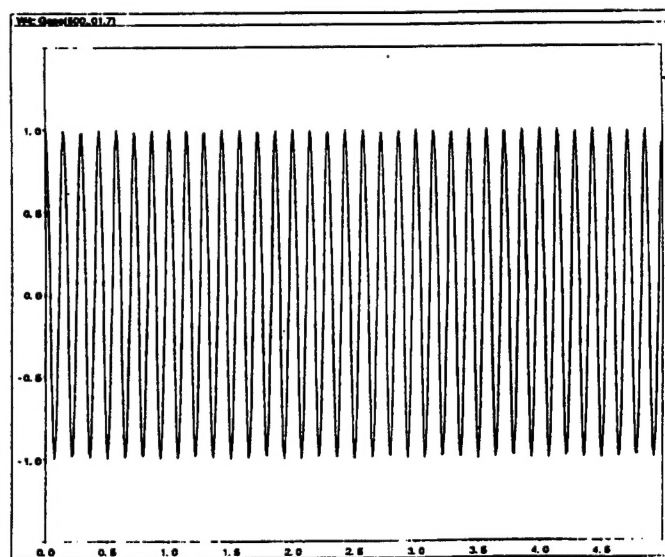
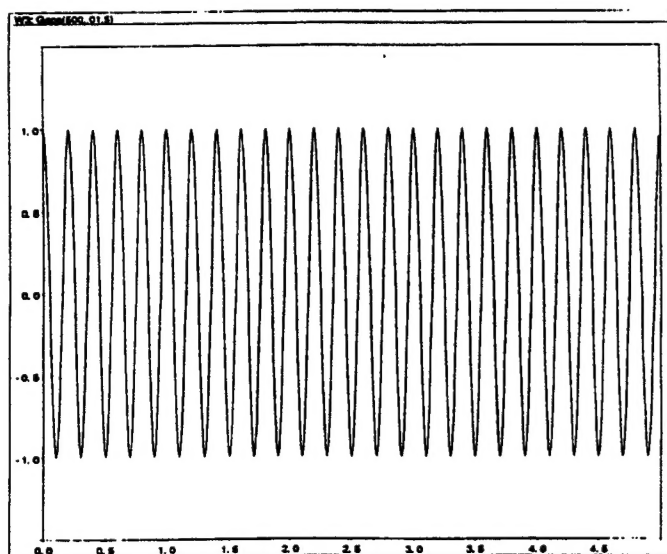
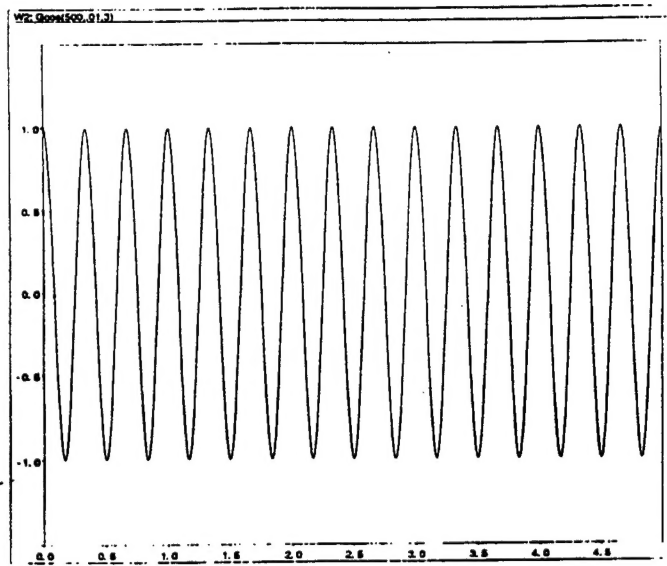
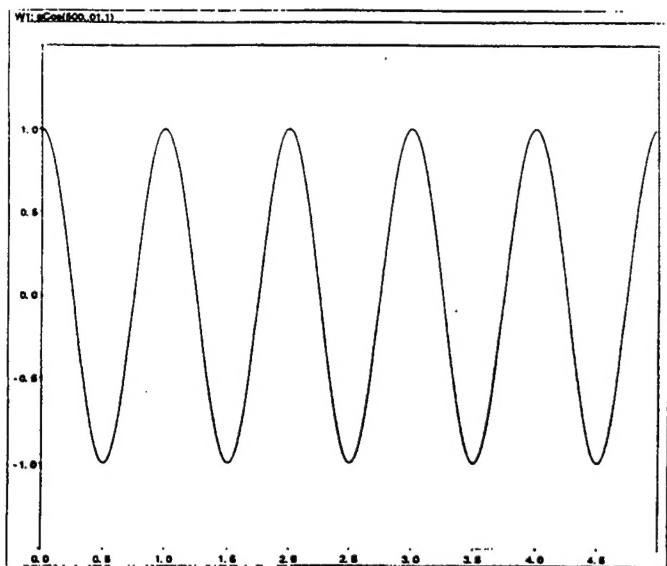
One of the first things that I did was to demonstrate, using software called DADiSP, how any waveform can be created from a combination of harmonically related sines and cosines. I saw this by alternately subtracting and adding a large number of odd harmonics to create a square wave according to the formula,

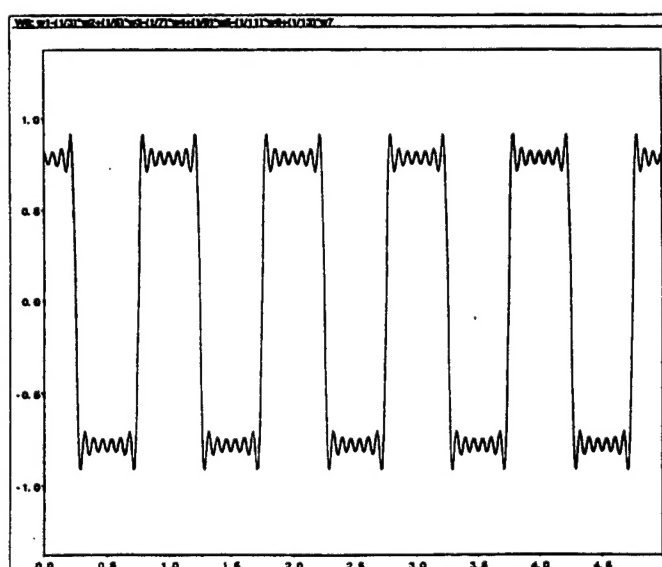
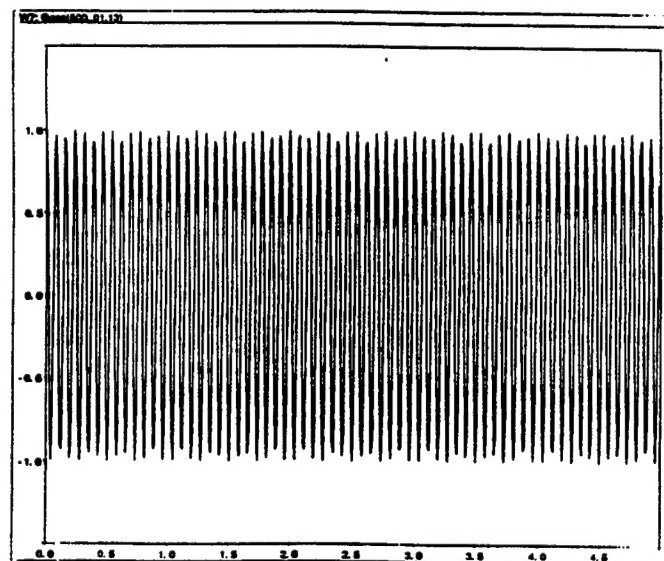
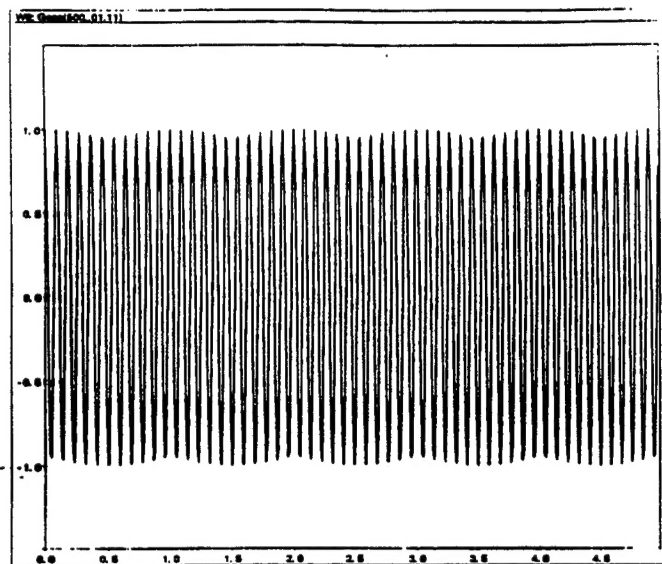
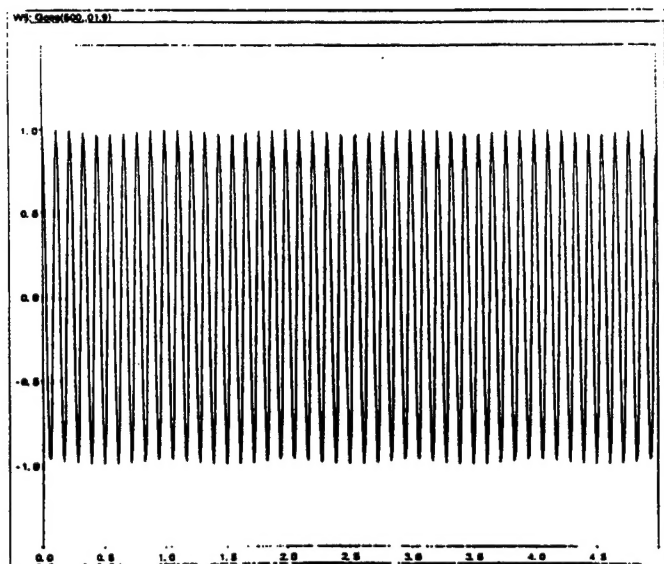
$$\text{square wave} = \sum_{\substack{n=1 \\ \text{odd}}}^{\infty} (1/n) \cos(2\pi n)$$

This formula will then give the equation,

$$\text{square wave} = \cos(2\pi) - (1/3)\cos(2\pi 3) + (1/5)\cos(2\pi 5) - (1/7)\cos(2\pi 7) + (1/9)\cos(2\pi 9)$$

After these calculations, the subtractions and additions of cosine wave forms will generate a wave which represents a square wave. This is demonstrated on the eight graphs on the following two pages.





The next thing that I studied during my tour was AM modulation. Modulation is defined as the process of transforming information from its original form to a form that is more suitable for transmission between a transmitter and a receiver. Demodulation is therefore the reverse process, meaning the signal is converted back to its original form. Amplitude modulation is defined as the process of changing the amplitude of a high frequency carrier in accordance with a modulating signal. Any frequency which can be transmitted through air and then received is known as RF which is the abbreviation for radio frequencies. The actual frequencies which have the signal you will actually here are known as the sidebands and are transmitted by the use of a carrier wave. These signals modulate the carrier and form what is known as the envelope. To solve for the signal in a time domain mathematically, the following equation would be used,

$$s(t)_{AM} = (1+m(t))\cos(2\pi ft)$$

The part of the equation which is the modulation is the $m(t)$ (fig 1). The carrier would then be $\cos(2\pi ft)$ (fig 2).

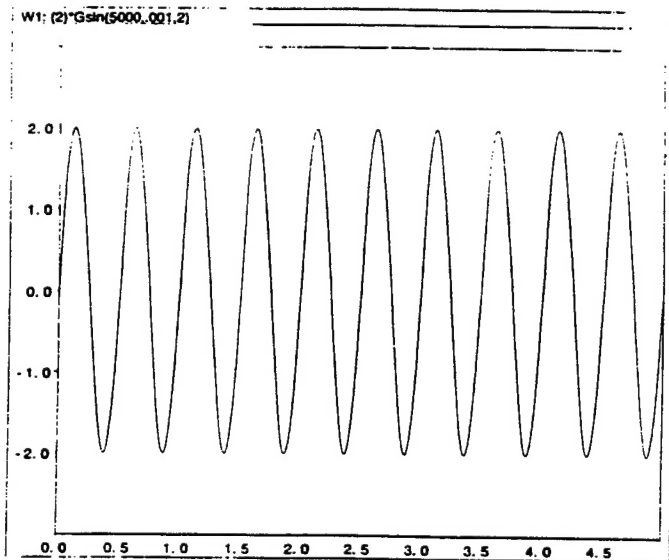


fig 1

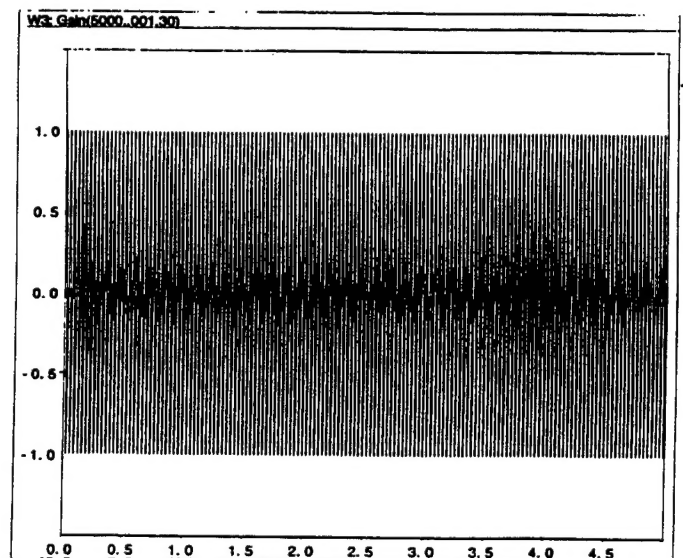


fig 2

After putting these two graphs through the equation, the result would be the graph of the signal in a time domain (fig 3). It can also be seen as a spectrum, which means viewing it in a frequency domain (fig 4). This AM modulation is called a double side band. It has a weak carrier and very strong side bands. Also, shown in figure 5 and figure 6 is the signal in a time domain and the spectrum of a signal with a weaker modulation and a weaker carrier wave.

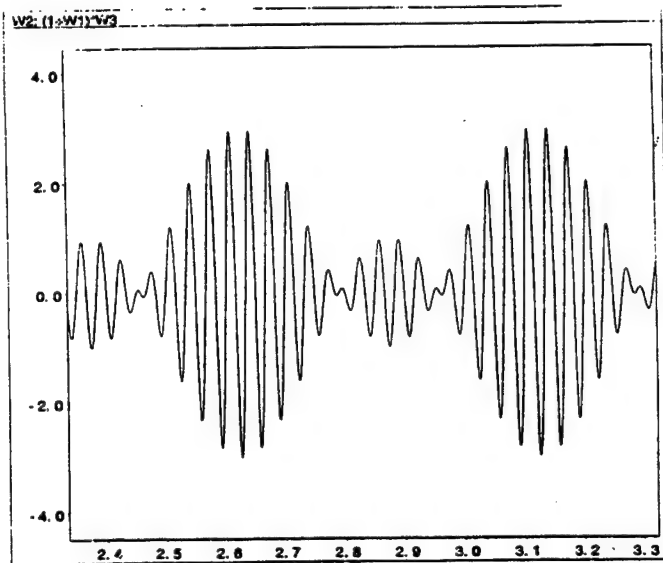


fig 3

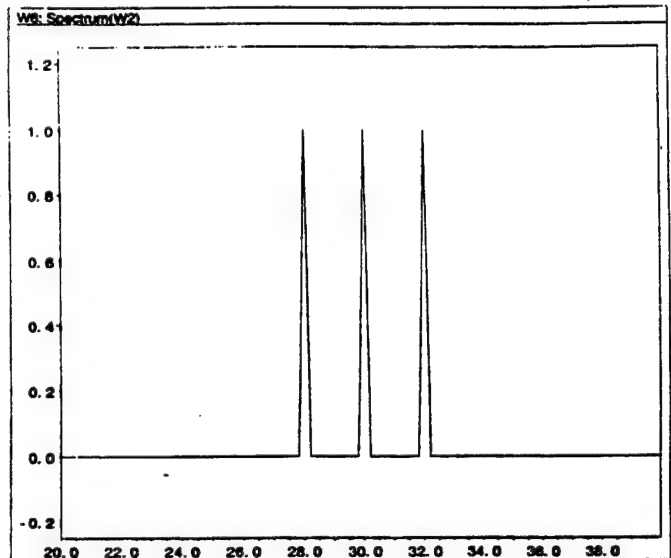


fig 4

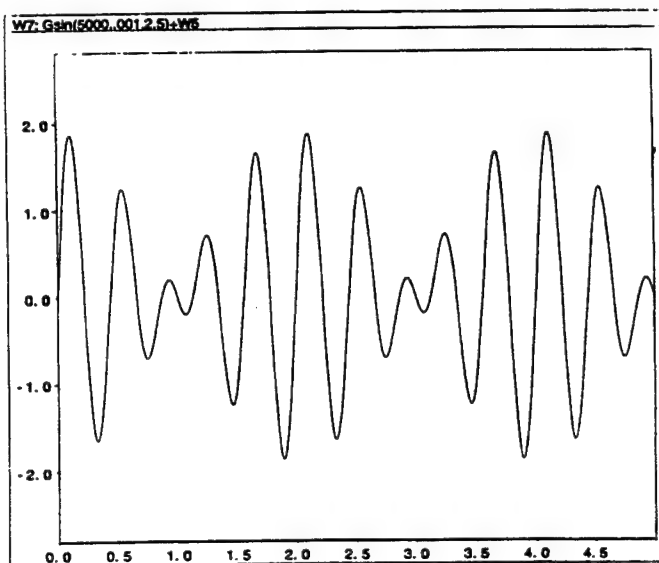


fig 5

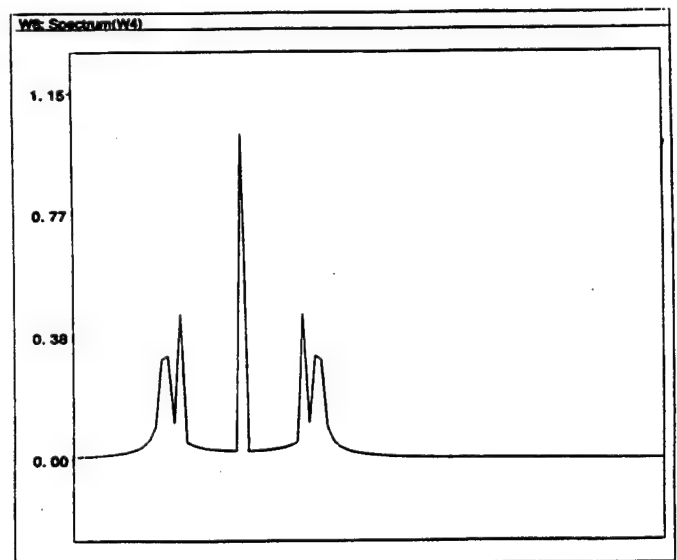
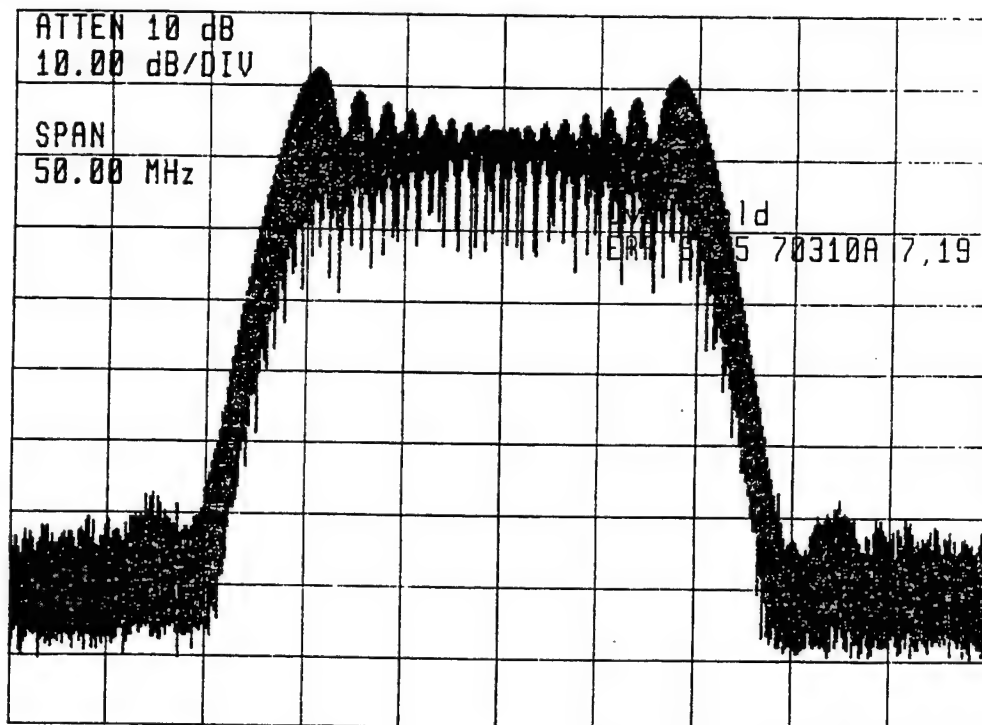


fig 6

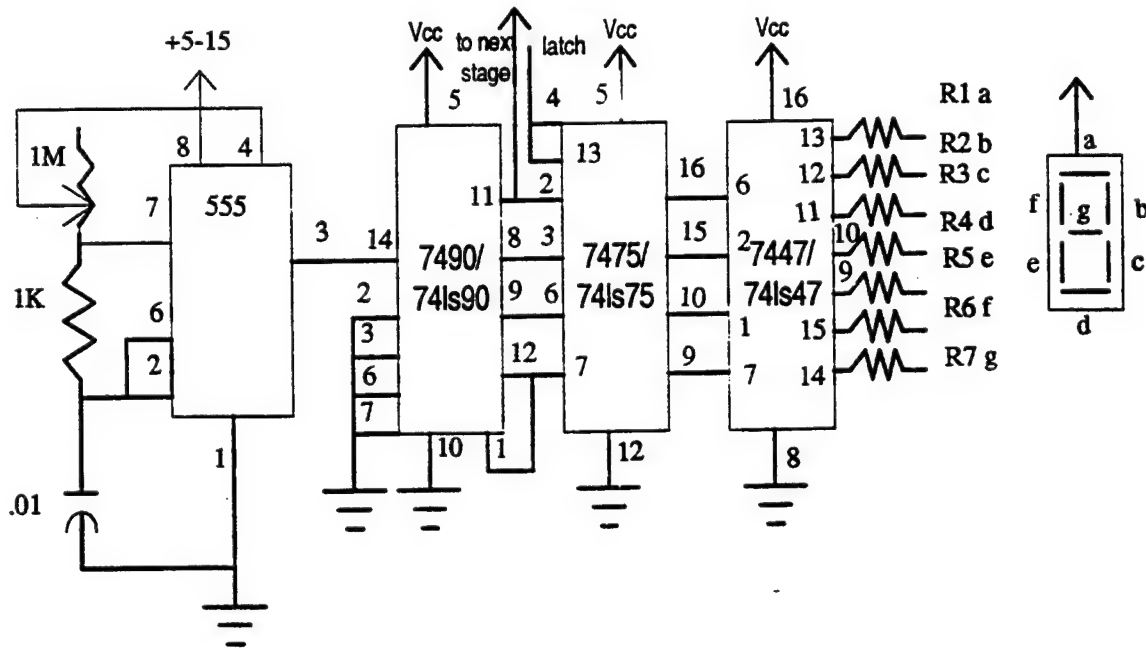
Another thing which was studied was FM modulation. Direct FM is defined as an angle modulation in which the frequency of the carrier varies directly by a varying modulating signal. Mathematically, an FM modulation can be solve by using the following equation,

$$s(t)_{\text{FM}} = A \cos(2\pi f + K \int v(t) dt)$$

This study was boosted by the new piece of equipment which we received at the beginning of my tour. This piece of equipment was a Hewlett Packard 8645A-Agile Signal Generator. This is a very powerful machine, and I was impressed with its versatility and ease of use. Being a new piece of equipment, I took it upon myself to test it out. I ran several of the examples in the manual, one of these being on FM modulation. For one example in particular, I hooked up the spectrum analyzer to the HP 8645A and set the center frequency, frequency span, and reference level to the specified specs. I then followed the rest of the instructions for entering frequencies into the HP 8645A and the spectrum analyzer showed an FM modulation of 600 MHz carrier with an FM deviation of 10 MHz and an audio frequency rate of 400 kHz. The plot below shows the spectrum analyzer display of a wideband FM signal with a modulation index of approximately 25.



Next during my tour, I studied circuits and, using breadboard, built a few of my own. One of the circuits I built was called a decimal counting unit. In order for this circuit to work, I had to build another circuit to supply a pulse, so I built a simple pulse generator. Following is a schematic of the completed circuit.



For this circuit, I built it with two stages. This way it was a two digit counter. To do this, I had to connect pin 11 on the first stage on the 7490 chip to pin 14 on the second stage on the 7490 chip. This counter will only count up to 99, then it will reset to 00. At first, it was counting so fast that it was impossible to see the numbers on the LED display. In order to slow it down, I tried to adjust the variable resistor on the pulse generator circuit. Even though it did slow down the pulses, it was still too fast. I then replaced the capacitor with a much larger one and that made a noticeable difference. Once this was accomplished, I wanted to be able to freeze the display momentarily to see the number. To do this, I placed a button on pins 4 and 13, labeled latch on the schematic, on the 7475 chip. This did not stop the pulse generator, but only froze the display while the button was held in. Once it was released, the display jumped to the digit the counter had reached while the button was held in.

Once I was finished with circuit building, I moved on to computer programming. I entered a few programs into Matlab on the Sun Sparkstation and a few into Mathcad. These programs simply solved for some mathematical operation and formed a graph. The first program my partner and I wrote in an actual program writing software was in GW Basic. We were rather discouraged with GW Basic because the files could not be read in a newer version of Basic such as HT Basic. That program was one which generated maximal length sequences. At the beginning of the program, the user inputs the number of digits that will be used to generate the sequence. This is called the length of the register. The user then inputs the initial fill of ones and zeros. Next, the user inputs the register taps. The program then displays the last number in the register after each clock pulse. During this, the program is checking to see if the register matches the initial fill condition and checks that it cycled the right number of clock pulses. If both requirements are met, the program tells you that it has reached a maximal length sequence.

The second major program that I wrote was in HT Basic. This program used the IEEE-488 interface to send data to an arbitrary waveform generator. The output of the waveform generator was a maximum length sequence for the polynomial x^4+x+1 . This sequence is 111010110010001. Following is the program itself and the plot of what was on the oscilloscope.

```

10 ! This program uses the arbitrary waveform function to
20 ! download and output a maximum length sequence
30 ! for the polynomial  $x^4+x+1$ . The sequence
40 ! consists of 3750 points downloaded to the
50 ! function generator as ASCII data. Written by
60 ! Jason Leopold on 28 July 1995.
65 !
70 OPTION BASE 1
80 DIM Waveform(3750)
90 INTEGER I
100 INTEGER J

```

```
110 A=1
120 B=250
130 DIM Seq(16)
140 Seq(1)=1
150 Seq(2)=1
160 Seq(3)=1
170 Seq(4)=0
180 Seq(5)=1
190 Seq(6)=0
200 Seq(7)=1
210 Seq(8)=1
220 Seq(9)=0
230 Seq(10)=0
240 Seq(11)=1
250 Seq(12)=0
260 Seq(13)=0
270 Seq(14)=0
280 Seq(15)=1
290 !
300 CLEAR 7
310 ASSIGN @Fgen TO 710
320 OUTPUT @Fgen;"*RST"
330 !
340 FOR K=1 TO 15
350 IF Seq(K)=1 THEN GOSUB 570
360 IF Seq(K)=0 THEN GOSUB 630
370 !
```

```

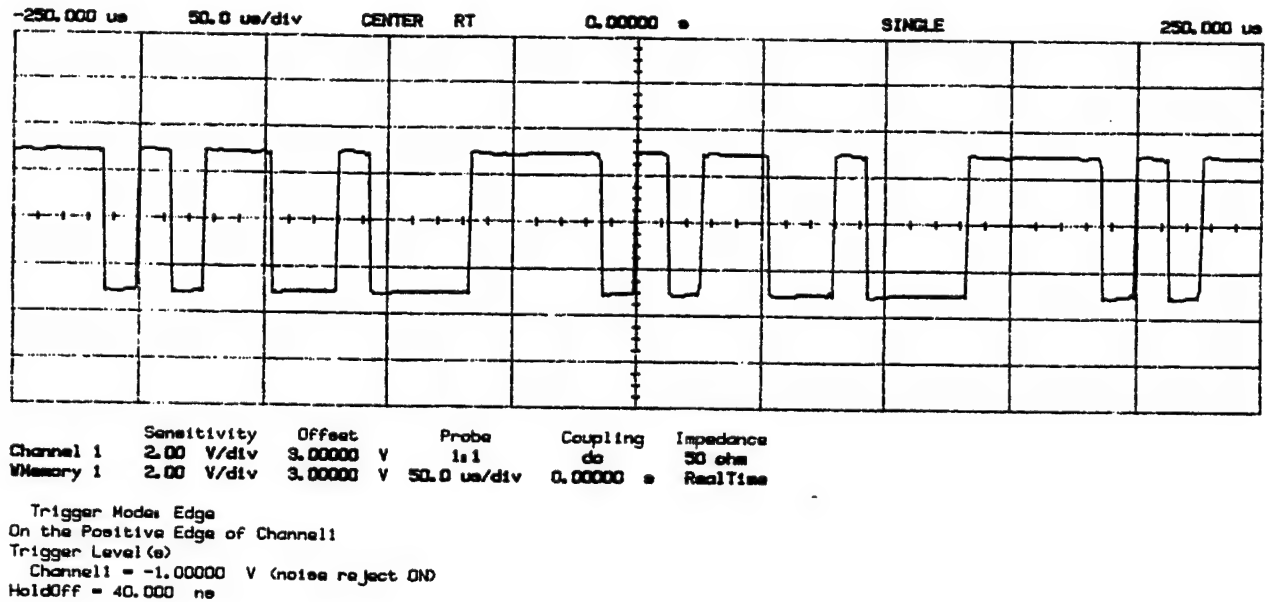
380 A=A+250
390 B=B+250
400 NEXT K
410 !
420 ! Download data points to volatile memory from array
430 !
440 DISP "Downloading Arb..."
450 OUTPUT @Fgen;"DATA VOLATILE,";Waveform(*)
460 DISP "Download complete"
470 !
480 OUTPUT @Fgen;"DATA:COPY PULSE, VOLATILE"
490 OUTPUT @Fgen;"FUNC:USER PULSE"
500 OUTPUT @Fgen;"FUNC:SHAP USER"
510 !
520 OUTPUT @Fgen;"OUTP:LOAD 50"
530 OUTPUT @Fgen;"FREQ 5000;VOLT 5"
540 !
550 GOTO 1000
560 !
570 !
580 FOR I=A TO B
590 Waveform(I)=1
600 NEXT I
610 RETURN
620 !
630 !
640 FOR I=A TO B

```

```

650 Waveform(I)=0
660 NEXT I
670 RETURN
1000 END

```



This plot shows the ones of the sequence as the top lines and the zeros are represented by the bottom lines. This plot is generated by sending 3750 points to the waveform generator. Each bit in the sequence is tested to see if it is a one or a zero. According to that, the program is sent to the appropriate line which sends a message to the waveform generator telling it what kind of waveform to output. This program could later be modified to plot the sequence for any give polynomial. To do this, the program previously discussed which generated maximal length sequences would have to be inserted. The program itself would also need many other modifications.

After studying computer programming, I moved on to working with the ADSP-21020 EZ-LAB Evaluation Board. This board is a programmable digital signal processor. The ADSP board is interfaced with a computer using an RS-232 cable connected to the COM port on the back of the computer. Using

the specific software, programs can be downloaded to the ADSP board which will generate many different effects. The three we experimented with the most were called talkthru, echo, and loop. At first, all of these programs worked for us. But, halfway through my tour, we got a new computer and installed the software on it. For some reason, we could only get the loop program to work and the one that was supposed to echo was working like talkthru used to work. However, the loop program had some very interesting functions. To operate the board after the program was downloaded was done only by three buttons on the ADSP. To record something that you wanted to loop, you simply depressed the button for the length of the segment of sound. The segment could only be approximately five seconds long. After the button is released, the ADSP board started looping through the recording. Pressing the second button would then play the recorded segment backwards. Pressing the second button again plays the segment forwards, but alters all the frequencies to a higher range. If the second button was pressed once more, the segment would be played normally. The third button would reset the board. We found that to fully reset the board, it had to be totally powered down and then powered up again. The only thing the reset button would really do was to reset the memory for the sound which had been recorded.

Another project that we did was linking two computers so that the drives on both computers could be used on one. I found it very interesting that this could be done. We used the software package called Comm Works to link the computers. Doing so made it very easy to transfer software from one computer to another. The nice thing about using Comm Works was that even when the two computers were linked, they could both still be used as their own operating system.

I feel that my apprenticeship here at Wright Patt was an excellent learning experience for me. It has really boosted my interest in Electronics Engineering. Along with the things discussed in this report, I received some good background on electronic circuits. I've learned how to troubleshoot and how to exchange components to alter the effect. After my tour is completed, I plan to spend some more time working with electronics and building a few circuits such as amplifiers and distortion pedals for electric guitars. I also plan to further my education to become an electronics engineer.

References

1. Wayne Tomasi, Electronic Communications Systems-Fundamentals Through Advanced, Prentice-Hall Inc., 1988.
2. Forrest M. Mims, III, Engineer's Notebook II-A Handbook of Integrated Circuit Applications, Radio Shack, 1982.

**INJECTION OF A LIQUID JET INTO A
SUBSONIC CROSS FLOW**

Scott A. McAlpine

**Centerville High School
500 East Franklin Street
Centerville, OH 45459**

**Final Report for the
High School Apprenticeship Program**

**Sponsored by Wright Laboratory,
Wright Patterson AFB**

and the

**Air Force Office of Scientific Research,
Bolling AFB, Washington DC**

August 1995

INJECTION OF A LIQUID JET INTO A SUBSONIC CROSS FLOW

Scott A. McAlpine
Centerville High School

ABSTRACT

The transverse injection of a water jet into a high speed cross flow was studied. The effects of air speed, injection rate, and nozzle diameter were the focus of the research. The study showed that jet penetration increases as momentum ratio increases. Penetration also increases as the injector diameter increases when the momentum ratio is held constant. The study also showed that the breakup and atomization processes are more effective with higher air speed, and that injection rate has little or no effect on the atomization process.

INJECTION OF A LIQUID JET INTO A SUBSONIC CROSS FLOW

Scott A. McAlpine

INTRODUCTION

Transverse injection of liquid fuels into high speed cross flows has many engineering uses, including transpiration cooling, thrust-vector control of rockets, and fuel injection in various types of jets. In particular, Interest in hypersonic hydrocarbon fueled missiles has been increasing rapidly in the last few years. This new interest has caused scientists to closely examine the behavior of liquid jets in order to design and develop ramburners that can operate at speeds of Mach 4-7 and at both sea level and high altitude. The designer of such an engine will need information of the tradeoffs between several important operating parameters, such as vehicle size and geometry, burner-area ratio, thrust capability, and specific impulse. As the design effort progresses, precise data will be needed on the specific features and operational characteristics of the combustion system, such as the configuration and location of fuel-injection sites and flame holders, the magnitude of heat transfer to combustor walls, and the composition of combustion products which determines nozzle efficiency. The designer must also have a detailed knowledge of the properties of the fuel in use, such as its cooling capacity, its atomization process, and how it may change under extreme temperatures and pressures {1}.

Investigation of the injection and atomization process in combustion systems which use high-density hydrocarbon fuels, especially systems which utilize endothermic fuels, will prove to be very important in achieving hypersonic flight. High-density, endothermic hydrocarbon fuels are important because they offer great potential for a more efficient cooling system for the combustor and airframe by undergoing chemical decomposition and/or thermal cracking before the actual injection takes place. To date, the properties of transverse fuel injection have been characterized by the properties of penetration height, droplet size, and mass-flux distribution. Lately, most engineers have focused on photographically studying the spray properties of the injectant.

A. Sherman and J. A. Schetz {2} were among the first engineers to do a detailed study of the structure of transverse liquid jets in high-speed flows by using photographic techniques. They determined that jet breakup is brought about by a cross-jet fracture at a wave trough. Joshi *et al.* {4} studied the effects of injector shape on the structure and penetration of liquid jets in a high speed cross flow. They found a correlation between penetration and jet to free stream pressure ratio. They also discovered that the ratio of frontal-to-streamwise dimension is affected by the shape of the injector. They discovered that a rectangular injector allows the highest penetration and spread. Nejad and Schetz {5} studied the effects of viscosity and surface tension on transverse injection. They found that droplet diameter and size distribution are inversely proportional to surface tension. Nejad and Schetz also discovered that a jet breaks up into ligaments rather than droplets for a certain range of viscosities and dynamic pressure ratios.

This paper contains the results of a further study of the effects of cross flow speed, injector size, and rate of injection on the processes of injectant penetration, breakup, and atomization.

EXPERIMENTAL FACILITY

A. INJECTION TUNNEL

A schematic diagram of the injection tunnel is shown in Figure 1. The tunnel was constructed to allow flow visualization and other optical diagnostics. The test facility consisted of a cylindrical settling chamber, followed by a 5-inch-wide x 3-inch-high x 48-inch-long transition section. A honeycomb-shaped flow straightener was added in this section in order to insure a uniform, two-dimensional flow. A screen was added in order to free the air of dust and other debris. The dimensions of the test section were the same as the dimensions of the settling chamber. The top and side walls were fitted with optical-quality quartz to allow for optical observation. High-pressure, high-capacity pumps provided a continuous air supply. The pumps were capable of providing a 30-lbm/s flow rate at 750 psi. The airflow rate was

measured by an orifice flowmeter located upstream of the flow control valve. A two-dimensional variable-area sonic nozzle located downstream of the test section allowed exact control of Mach number in the test section. Measurements of temperature and pressure were made at several locations to allow monitoring and control of the air flow. A personal computer based data-acquisition system monitored, recorded, and calculated important flow characteristics, such as Mach number, air temperature, air pressure, air mass flow rate, and injectant mass flow rate. The variable-area sonic nozzle was connected to an exhaust system which releases the air into the atmosphere. A 286 microcomputer was used to monitor the air pressure, air temperature, Mach number, and mass flow rate.

B. FUEL-INJECTION SYSTEM

The fuel-injection system consisted of a pressurized fuel tank, a flow-control valve, a positive-displacement flowmeter, and two interchangeable round orifice injectors. The fuel (water) tank was pressurized with nitrogen at 300 psi. A fast-acting pressure regulator was used to insure a constant fuel tank pressure in order to maintain a steady water flow during the tests. The water mass flow rate was measured with a positive-displacement flowmeter to allow precise calculation of the fuel-to-air ratio. Two circular-port brass injectors (.02 and .04 inches in diameter) were mounted flush with the bottom wall of the test section.

C. OPTICAL-DIAGNOSTIC SYSTEM

The optical configuration for observing the water jet is represented by Figure 2. An Nd: YAG laser was used as a light source. It was set to emit a single pulse of light with a duration of 8 ns. The wavelength was set at 5320 Å. A series of conventional lenses and mirrors were used to collimate the beam to a diameter of two inches while it traveled through the test section. Another lens was used to refocus the beam into a Polaroid Graphic Speed camera. Photographs were taken on 4" x 5" Polaroid type 52 film. The magnification factor for this experiment was 2.50.

RESULTS

A. JET BREAKUP

The Breakup process:

The typical stages of liquid jet breakup are shown in Figure 3. The breakup of a liquid jet can be divided into three stages. The first is the liquid column stage. In this stage, the jet is completely intact with only small deformations. During this stage, the jet begins to undergo two types of breakup: column breakup and shear breakup. Column breakup is apparently the result of instabilities in the airflow and in the liquid jet after the air and liquid interact with each other. The liquid begins to vibrate because of these instabilities, and then it begins to break up. Shear breakup is caused by the airflow "stripping" droplets off of the water as the air goes around the jet. Shear breakup begins to take place almost immediately after injection, and is more evident when the injection rate is higher because the liquid column exists for a longer period of time, and there is more time for the shear breakup to occur. The second stage of the breakup process is the ligament stage. During this stage, the liquid exists in large patches. Further instability in the liquid and air cause the ligaments to break up into small droplets and individual molecules. This is the third stage of jet breakup. The liquid is now considered to be atomized and does not break up any further.

Factors affecting breakup:

The study showed that jet breakup is greatly affected by air speed and by injector diameter. It was observed that the breakup occurred much more quickly with a relatively high Mach number than with a relatively low Mach number. This is due to the fact that the faster air exerts a greater force on the water jet, and therefore breaks it up more quickly than slower air would. The second factor that was observed to affect jet breakup is injector diameter. As the diameter increases, breakup occurs at a slower rate. This is simply due to the fact that there is more matter for the air to break up. It takes the air a longer period of time to break up a large amount of matter than it takes to break up a small amount of matter. The study showed that breakup is affected little, if at all, by injection rate.

B. JET PENETRATION

Jet penetration was affected by two variables: air speed and injection rate. As air speed increased, penetration decreased; and as injection rate increased, so did penetration. These two factors compete against each other in order to determine how far the jet penetrates. Injection rate and air speed are commonly related in a quantity called momentum ratio (q), which is defined as:

$$q = \frac{\rho_f v_f^2}{\rho_g u_g^2}$$

where ρ_f is the fluid density, v_f is the fluid velocity, ρ_g is the air density, and u_g is the air velocity.

Since momentum ratio is defined as injectant momentum divided by cross flow momentum, an increase in injectant momentum will result in a higher momentum ratio. As the momentum ratio between the air and injectant increases, the injectant will penetrate further before atomization. This is due to the fact that the momentum of the air and the momentum of the liquid compete against each other.

The study also showed that when the momentum ratio is kept the same and the injector diameter is increased, the penetration increases. This can be explained by the fact that the larger injector diameter causes more water to be sent into the cross flow. This gives the water much more inertia than it has with a smaller injector. This inertia increase causes the jet to maintain its upward velocity for a longer period of time, and therefore it penetrates further.

CONCLUSIONS

The following conclusions were drawn from this study:

1. A liquid jet injected into a cross flow has three main stages--the liquid column stage, the ligament stage, and the droplet (atomized) stage.
2. Jet penetration increases as momentum ratio increases. Penetration also increases as injector diameter increases if the momentum ratio is kept the same.
3. The breakup of a liquid jet occurs at a quicker rate when air speed is increased, and it occurs at a slower rate when the injector diameter is increased. Breakup is not affected significantly by injection rate.

REFERENCES

1. Chen, T.H., Smith, C.R., Schommer, D.G., and Nejad, A.S. "Multi-zone Behavior of Transverse Liquid Jet in High-Speed Flow," AIAA Paper 93-0453. Jan 1993.
2. Sherman, A., and Schetz, J. A., "Breakup of Liquid Sheets and Jets in a Supersonic Gas Stream," *AIAA Journal*, Vol. 9, No. 4, April 1971, pp. 666-673.
3. Kush, E. A., and Schetz, J. A., "Liquid Jet Injection into a Supersonic Flow," *AIAA Journal*, Vol. 11, No. 9, September 1973, pp. 1223-1224.
4. Joshi, P., and Schetz, J. A., "Effects of Injector Shape on Penetration and Spread of Liquid Jets," *AIAA Journal*, Vol. 13, No. 9, September 1975, pp. 1137-1138.
5. Nejad, A. S., and Schetz, J. A., "Effects of Properties and Locations in the Plume on Droplet Diameter for Injection in a Supersonic Stream," *AIAA Journal*, Vol. 22, No. 4, April 1984, pp. 653-659.

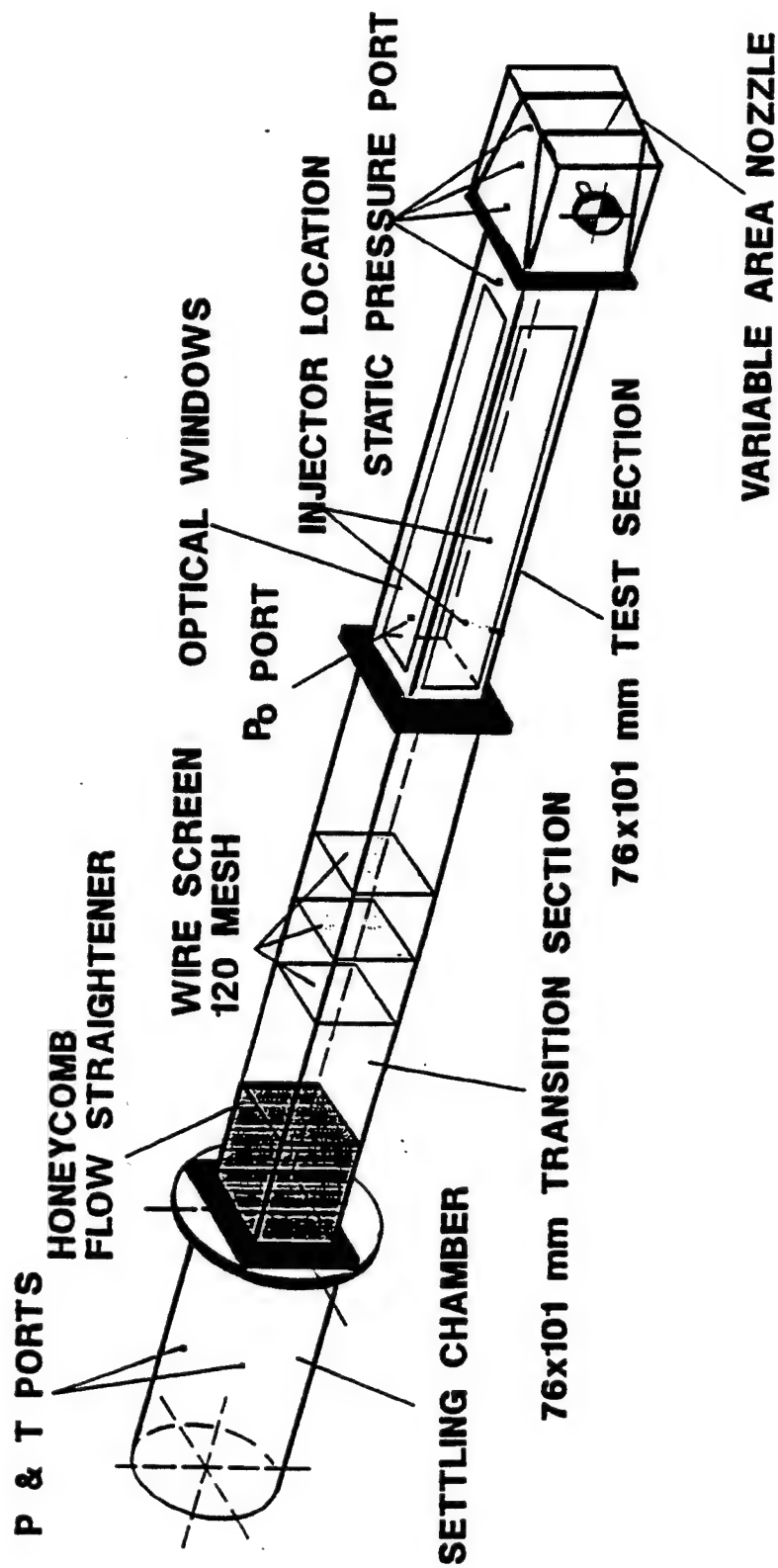


Fig.1--Diagram of the injection tunnel.

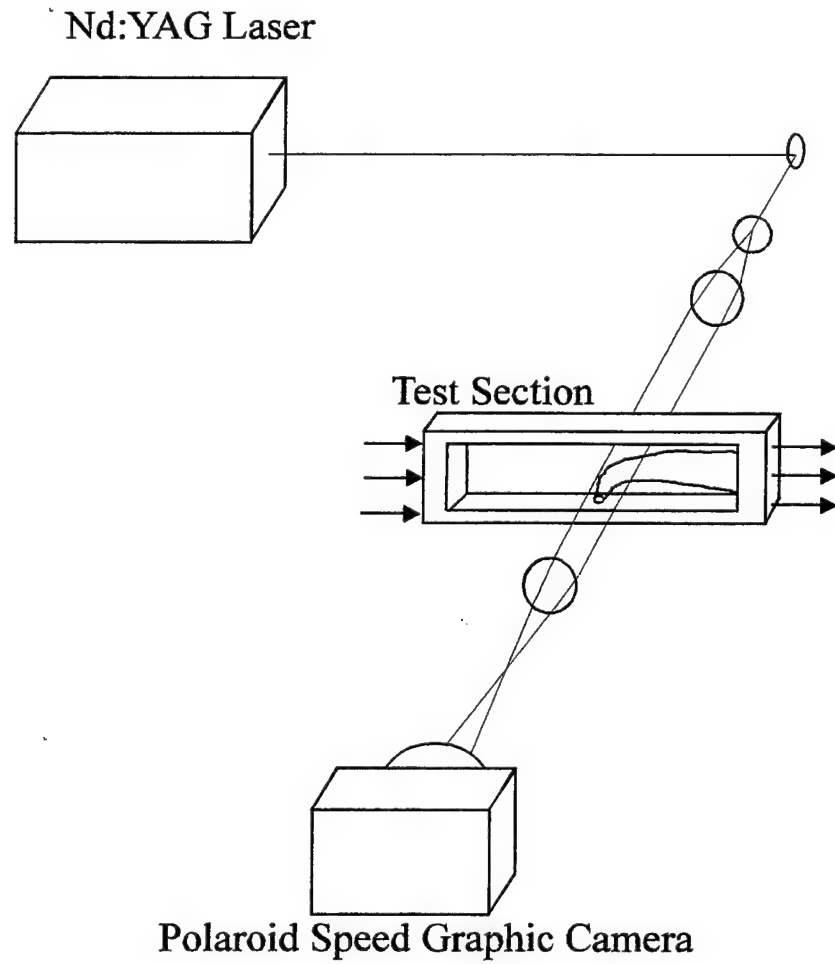


Fig 2--Sketch of optical diagnostic system.

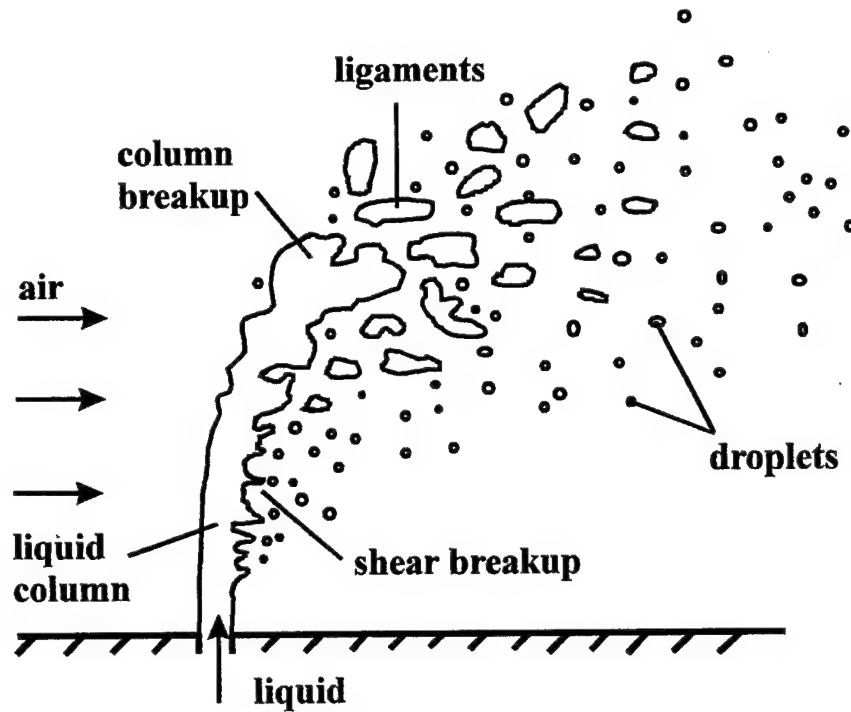


Fig 3a--A sketch of liquid jet breakup.



Fig 3b--Photograph of liquid jet breakup

An Attempted Study of the Minimum Deviation Method
for Refractive Index Measurements

Sandra R. McPherson

Bishop Brossart High School
Jefferson & Grove Sts.
Alexandria, KY 41001

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

AN ATTEMPTED STUDY OF THE MINIMUM DEVIATION METHOD
FOR REFRACTIVE INDEX MEASUREMENTS

Sandra R. McPherson
Bishop Brossart High School

Abstract

The index of refraction, which is a measure of how light travels through a substance, is a basic characterization done by scientists after a new material is created. The purpose of this study was to determine the effect of temperature on the index of refraction. Due to unforeseen circumstances, this was not possible.

AN ATTEMPTED STUDY OF THE MINIMUM DEVIATION METHOD
FOR REFRACTIVE INDEX MEASUREMENTS

Sandra R. McPherson

The purpose of my summer in the AFOSR High School Apprentice Program was to study the index of refraction as in the two previous summers. This summer, instead of measuring the index at room temperature, the goal was for me to find the index at a variety of higher temperatures as well as at a variety of wavelengths. Not only would this tell us more about the crystal, it would hopefully improve accuracy too. If the room temperature did not fluctuate during the measurements then the index would remain constant and hopefully more precise. This was to be accomplished by replacing the normal stage used with the Gaertner spectrometer with a heated stage designed and built by Paul Von Richter of the University of Dayton Research Institute.

When I started my summer term the stage was not yet complete. A problem arose in trying to stabilize the crystal on the stage so it would not move during measurements. This was solved by using springs to hold down the crystal while not interfering with the light as it passed through. In the meantime, I refreshed my memory and refined my measuring technique using crystals with known indices. These measurements were not important unless I could test them over a range of frequencies. I then had to leave my job for a period of time to participate in the Kentucky Governor's Scholar Program. While I was gone, the heated stage was completed. Unfortunately, that stage was used and broken by someone else in my lab attempting to study the index of refraction before I could return. The stage was not able to be fixed for the remainder of my term.

I tried to spend my free time wisely by assisting with other experiments going on in the lab at the time. I aided in the growth and harvesting of nonlinear crystals and I spent much time in the technical library attempting to find other ways to measure the index of refraction in the 5-10 micron range. On the day I completed my tour, I was able to find several light sources with output in that range and I alerted the lab to a section on infrared detectors. Hopefully my work will not be in vain and a method will be developed soon.

In conclusion, I would like to thank the AFOSR and RDL for the opportunities given to me through my three summers as a high school apprentice. I enjoyed participating in this unique program and I also enjoyed the experience of working daily in such an impressive place as Wright Patterson Air Force Base.

PROGRAMMING OF PHASED MICROMIRROR
ARRAYS AND IMAGE TRUTHING
OF TARGET CUER TESTING

Angela R. Miller

Wayne High School
5400 Chambersburg Road
Huber Heights, OH 45424

Final Report for:
High School Apprentice Program

Sponsored by:
Wright Laboratory
and
Air Force Office of Scientific Research
Bolling Air Force Base
Washington, DC

September, 1995

PROGRAMMING OF PHASED MICROMIRROR
ARRAYS AND IMAGE TRUTHING
OF TARGET CUER TESTING

Angela R. Miller
Wayne High School

Abstract

Micromirrors were devised into a computer program in order to steer a laser beam into a specific direction. The program allows for greater accuracy and efficiency within the beam steering itself. This devised program converts the mathematical output into pictures. These pictures, then allow for the data to be seen in order to tell if the correct outcome was projected.

Infrared images were taken from F-15E fighter jets, so that they could be truthed to allow pilots to test their accuracy in identifying targets correctly. The images were used for testing the automatic target cuers. Truthing is done in many steps, but one was only done here.

PROGRAMMING OF PHASED MICROMIRROR ARRAYS AND IMAGE TRUTHING OF TARGET TESTING

Angela R. Miller

Two different areas were worked on this summer involving electro-optical systems. Beam steering and target analysis both happened to produce advanced experience with computers. An expected outcome program for a computer had to be produced ,and test flight imagery was to be truthed.

Beam steering can use phased arrays in order to cause a beam shift. To get this phase shift, micromirrors had to be developed. Micromirrors can be used very easily because of their size. Being this small, in microns, they work tremendously well on aircraft. For micromirrors to be used in beam steering, a program had to be devised to come up with the expected outcome. Mathematica, an existing program, was used to get this information. Model displays also had to be set up in order to see the outcome in picture form. Symbols were developed in a computer language in order to get the pictures from the model display. A single micromirror image was developed , but then it was replicated hundreds of times. The program would convert this data into the expected outcome of the phased shifts. Later testing will be done once the micromirrors are actually built. Then the computer expected outcome can be put to test against the "real" data.

The second task was image truthing. Truthing was the first step in the process of testing automatic target cuers (ATCs) with imagery collected by Air Force sensors. ATCs were developed to reduce exposure time of aircraft to combat rich areas. Precise images had to be taken in order for the pilots to get an accurate reading on the target.

Truthing is one way that pilots can accurately be tested on their ability to accurately identify a target. An air and ground truth must be taken for image truthing to occur. An "air truth" is the actual digitized imagery and the aircraft coordinates. A "ground truth" is the location of the targets, operating state of the targets, and the environment conditions. Each test included about 3800 frames, and each set of 29

frames was called a file. An overlay had to be placed on each individual frame picture. This had to be done either by manual placement or interpolation. These overlays were then used to show the actual target position and orientation on the image.

The overlays were used to test pilots in their accuracy by using the information that the pilot gave and if it matched to the information given on the overlay itself. This would then help the pilot realize how many enemies they could recognize accurately.

By working with these two different concepts, I was able to learn why computers are so important in today's work force. If I would have had to do all the set up by hand, my jobs would have taken years instead of a couple of weeks. Even though it was hard to completely understand the concepts behind the work, I was still able to learn why it had to be done. Hopefully, the information I was able to contribute will help the scientists that continue to work very hard on this project.

Eight Week Tour at Wright Patterson Air Force Base-Flight Dynamics Directorate

Raviraj Nataraj

Beavercreek High School
2660 Dayton-Xenia Road
Beavercreek, Ohio 45434

Final Report for:
High School Apprenticeship Program
Wright Patterson Air Force Base
Flight Dynamics Directorate
Structures Division
Design Methods Development Section

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Patterson Air Force Base

August 1995

Eight Week Tour at Wright Patterson Air Force Base-Flight Dynamics Directorate

Raviraj Nataraj
Beavercreek High School

Abstract

The following report details the summer research accomplishments of Raviraj Nataraj at the Wright Patterson Air Force Base-Flight Dynamics Directorate as part of an eight week tour sponsored by the Air Force Office of Scientific Research and High School Apprenticeship Program.

Final Report

Eight Week Tour at Wright Patterson Air Force Base-Flight Dynamics Directorate

Raviraj Nataraj

I spent an eight-week summer tour at Wright Patterson Air Force Base (WPAFB) Flight Dynamics Directorate, Structures Division, and it proved to be an enjoyable, learning experience. The tour came about through the High School Apprenticeship Program(HSAP). The overall mission of the Structures Division is to develop airplane structures. This is accomplished by applying computers to calculate and process information needed to design aircraft components. I was assigned to work in the Design Methods Development Group of which Mr. Nelson D. Wolf is the technical manager. My assigned mentor for the tour was Dr. Vipperla B. Venkayya, and I stayed in room 122A which is the office of Mr. Duane Velely and Mr. Amar Bhungalia. I worked on a Personal Computer(PC). I had limited experience working with computers and no knowledge of the FORTRAN(FORMula TRANslation), programming language predominantly used here, prior to starting my tour. Dr. Venkayya allowed me to spend my first couple of weeks familiarizing myself with the basics of FORTRAN. I spent much time reading about FORTRAN and writing and running small programs. Mr. Velely was very helpful in answering questions regarding the computer and its applications such as Fortran Power Station, X-vision, and Microsoft Word. Dr. Venkayya also spent time with me to explain details concerning FORTRAN itself. I gained knowledge on FORTRAN fundamentals such as do loops, while loops, arrays, formatting, subprograms, variable types, and other basic commands. In addition to my first two weeks, I also had the opportunity to attend cadet tours organized by Mr. Bhungalia. The tours were geared towards exhibiting Structure Division's facilities and test capabilities. The fatigue testing being done included those which measured heat and sound stresses. An additional tour was also added which explored a laboratory for flight simulation. The flight simulator itself was large and exhibited solid realism in its simulation runs.

After an initial period, I began to work with larger programs. The program Analyze/Danalyze was developed by Dr. Venkayya and Maj. Robert Canfield. It calculates information for a structure based on input factors such as nodes, elements, boundaries, and materials. The program itself is widely distributed among universities such as Purdue. I used the program to calculate information for a ten bar truss. A ten bar truss is a structure composed of six gridpoints connected by ten bar elements. Using input data instructions for Analyze/Danalyze, I created a data file and computed results such as weight, stress, and displacements. After working with Analyze/Danalyze, I looked at the program ICE. ICE is the database for a large program called ASTROS(Automated STRuctural Optimization System). ICE was originally designed to operate on a UNIX operating system, and no version of ICE exists on a MS-DOS operating system. Therefore, my original task to run ICE on a 486 PC and locate errors was not possible because my computer training at the high school level was not adequate to tackle this job. However, I did learn a few rules about the debugging process. The work with ICE lead to the frequently used application of ASTROS itself. ASTROS processes the same type of information as Analyze/Danalyze, but ASTROS is much larger and comprehensive with features such as run-time summaries. To better understand ASTROS, I looked over the User/Programmer manual as a reference to better decipher some of the files designed for ASTROS. Universal Analytics Incorporated, the software contractor for ASTROS, sends WPAFB a new version of ASTROS every six months along with test files. My assignment was to run the 71 test files on ASTROS version 12, the latest installment, and then print and label the output files. These were needed to check the program output. To run the files, I was given my own account on the CONVEX mainframe which, unlike the PC that ran on MS-DOS, ran on a UNIX operating system, and I needed to learn a few basic commands and to apply an x-editor. Mr. Ed Forster showed me how to create files showing differences from the output files produced by ASTROS12 compared to the result files from Universal Analytics. The account I received on the CONVEX also proved useful in storing files produced by an application called Hyper-Mesh. Ms. Geetha Bharatram explained to me how to use Hyper-Mesh in order

to see and print computer images of an airplane produced by a data file sent to my account. The last item I dealt with on my tour was examining a sandwich structure. A sandwich structure is composed of a center core of material in a regular hexagonal pattern on the top and bottom plates. I labeled gridpoints on a diagram of the honeycomb panel and then observed a few relationships of the regular hexagonal pattern. I then studied a program devised by Mr. Bhungalia that outputted all the gridpoints of a sandwich structure based on inputted X,Y, and Z dimensions. In addition to the preceding activities, I also spent time practicing FORTRAN.

In conclusion, my eight week summer tour at WPAFB, Structures Division, Design Methods Development Section, was enjoyable and beneficial. The atmosphere was friendly and relaxing as all the people I encountered were kind, patient, helpful, and knowledgeable. Also, I was allowed to work and learn at my own pace. I gained helpful knowledge of computers, their applications, and programming. I am still uncertain how the tour may influence my future endeavors and employment interests, but the opportunity to consistently observe and be part of a working environment will be valuable experience. I would like to thank Dr. Venkayya, Mr. Wolf, and others for taking time to help me in various ways throughout the eight weeks. I would also like to thank WPAFB and HSAP coordinator Mr. Adolph Harris for allowing me to take part in this beneficial eight week summer program.

A STUDY OF POLYMER LUMINESCENCE

Ann Nee

Centerville High School
500 E. Franklin Street
Centerville, OH 45459

Final Report for:
High School Apprentice Program
Wright Laboratories

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratories

August 1995

A STUDY OF POLYMER LUMINESCENCE

Ann Nee
Centerville High School

Abstract

The luminescence properties of synthesized polymers were studied. Specifically, tests were run to determine the relationship, if any, of temperature and excitation wavelength, to intensity and peak luminescence wavelength. Initial results indicate that although lower temperatures improve the brightness of the luminescence, high temperatures draw the color nearer to the desired blue.

A STUDY OF POLYMER LUMINESCENCE

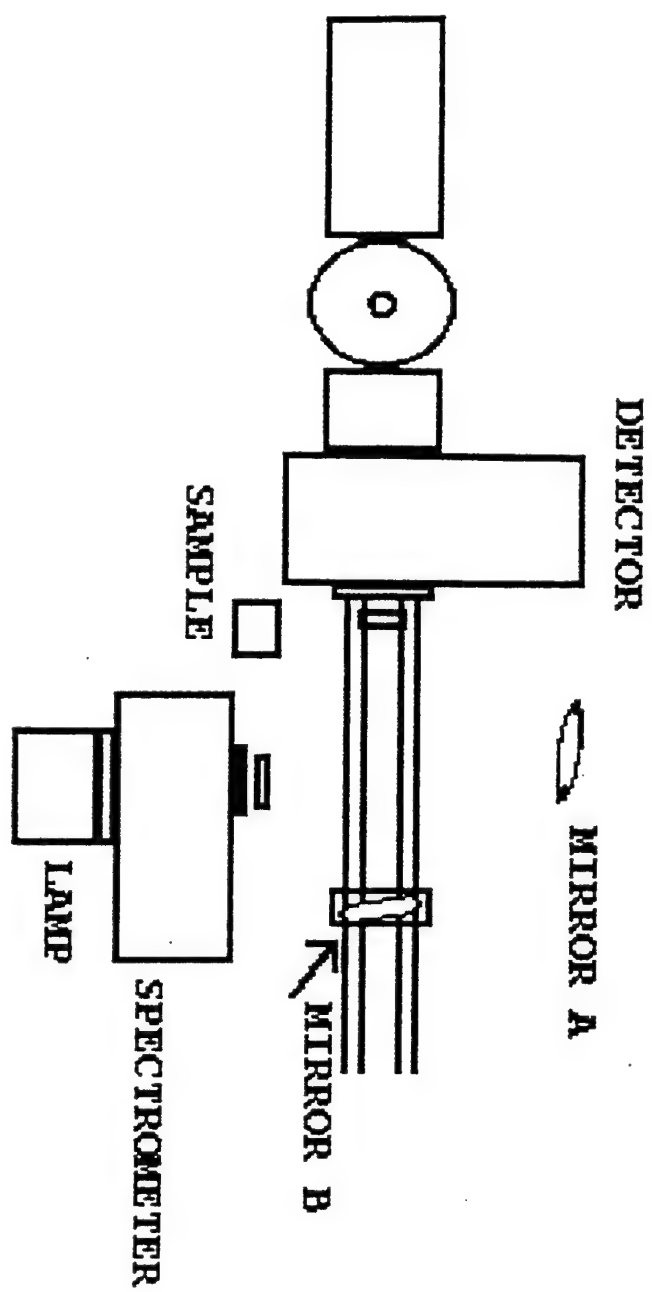
Ann Nee

Introduction

A few months prior to my arrival the polymer division of Wright Laboratories became interested in luminescent polymers, and the physics branch was set to test the products of the synthesis branch, with the possible benefit of developing efficient, reliable electroluminescent polymers. There are two main methods of creating luminescence--with light and with current. Both of these deliver an energy that moves the molecules to an excited state, and when they return to ground state, they luminesce. The difference in energies between the two states determines the spectrum, and it is possible to shift the spectra of a polymer by raising or lowering the band gap. It had been discovered previously that, if a polymer had both photoluminescence (PL) and electroluminescence (EL), the two spectra would often coincide; consequently, researchers have mainly concentrated on developing and testing a strong PL material, then exploring that polymer's EL capabilities.

Methodology

All the tests were set up as in the diagram on the next page. Free-standing polymer films were stapled into a cardboard frame, while thin films were mounted on glass, then placed in a small clamp for general PL and absorption tests, or in a cryostat for temperature studies. For the temperature studies, the samples were slid before the glass window into a small metal sleeve inside the cryostat, which was then put under vacuum. At this point, the center tube of the cryostat would be filled with liquid nitrogen, to aid in the cooling.



A connected temperature monitor would inform us as to when the sample had reached the target temperature; a few minutes of waiting would insure a uniform and stable distribution of the heat or cold, even in the slow-to-adapt cardboard. For alignment, we would first insure that the light coming from the monochromator and reflected off mirror A was focused on the sample; then, swivel and raise/lower mirror B to shine the resultant luminescence firmly into the slot on the detector. Often when using the cryostat, there would be several reflections (due to the fore glass window) between which one had to be chosen as the actual luminescence. Once this was done as well as could be by eye, test spectra were taken to determine how accurate the alignment was. Between each spectrum, we would twist one of the adjusting knobs on mirror B, deciding that when the spectrum was brightest and at the right wavelength, we would have the luminescent in the sharpest focus and best position possible. If the mirror was turned too far on its base (as when the original alignment was mostly inaccurate), however, large amounts of background would begin to appear that led to distortion of the curves. The spectra were generally displayed as counts versus wavelength for each of the 256 columns of 1024 sensors on the detector. Every so often a little helium-neon laser would be shone from the position of mirror A to help align the centers of the mirrors with the detecting slits. When finished, we turned off all sources of light except the lamp and draped the entire experiment area with black cloth to block any remaining reflections and to cover the gleams of the lamp from where it was mounted behind the spectrometer.

The tests were run for 64, 128, or 256 seconds, depending on the strength of the signal, with filters to block out both the excitation and the second order light. Since the backgrounds were also taken for the same amount of time, and the results from the subtraction of the relevant background from each test

were divided by the corresponding factor of 64, the differences in collection times were of negligible effect. In the absorption experiments we did manually, tests were run from 350 to 490 nm, at 10 nm intervals. In our temperature studies tests were run from -167 to 290° C, at roughly 15 to 20 degree intervals. Sometimes the data was taken before the cryostat reached the actual desired temperature; but since the heaters slowed further the closer they got to the stopping temperature, the environment was still stable enough to provide viable results.

The sample of PBT (poly(p-phenylene benzobisthiazole)) used for this round of experiments had been placed in an ethanol wash for two months. Ethanol had been chosen after some experimentation with other solutions (like methanol) had shown it to have the greatest effect in improving the spectrum.

In order to begin analysis, the data we had gathered had to be transformed into a system that Easy Plot and the rest of our Windows applications would recognize; we chose ASCII. After subtraction of the more outrageous cosmic spikes from the data, a program written by a coworker was employed to fit our data to Gaussian curves. The benefit of doing this, besides just having a mathematical way to characterize the curve, was that we could also learn the offset, energy, amplitude and width of each peak, and from these, the integrated area. All of this was then graphed in Easy Plot versus temperature (or 1000/temperature) to determine if there was a relationship.

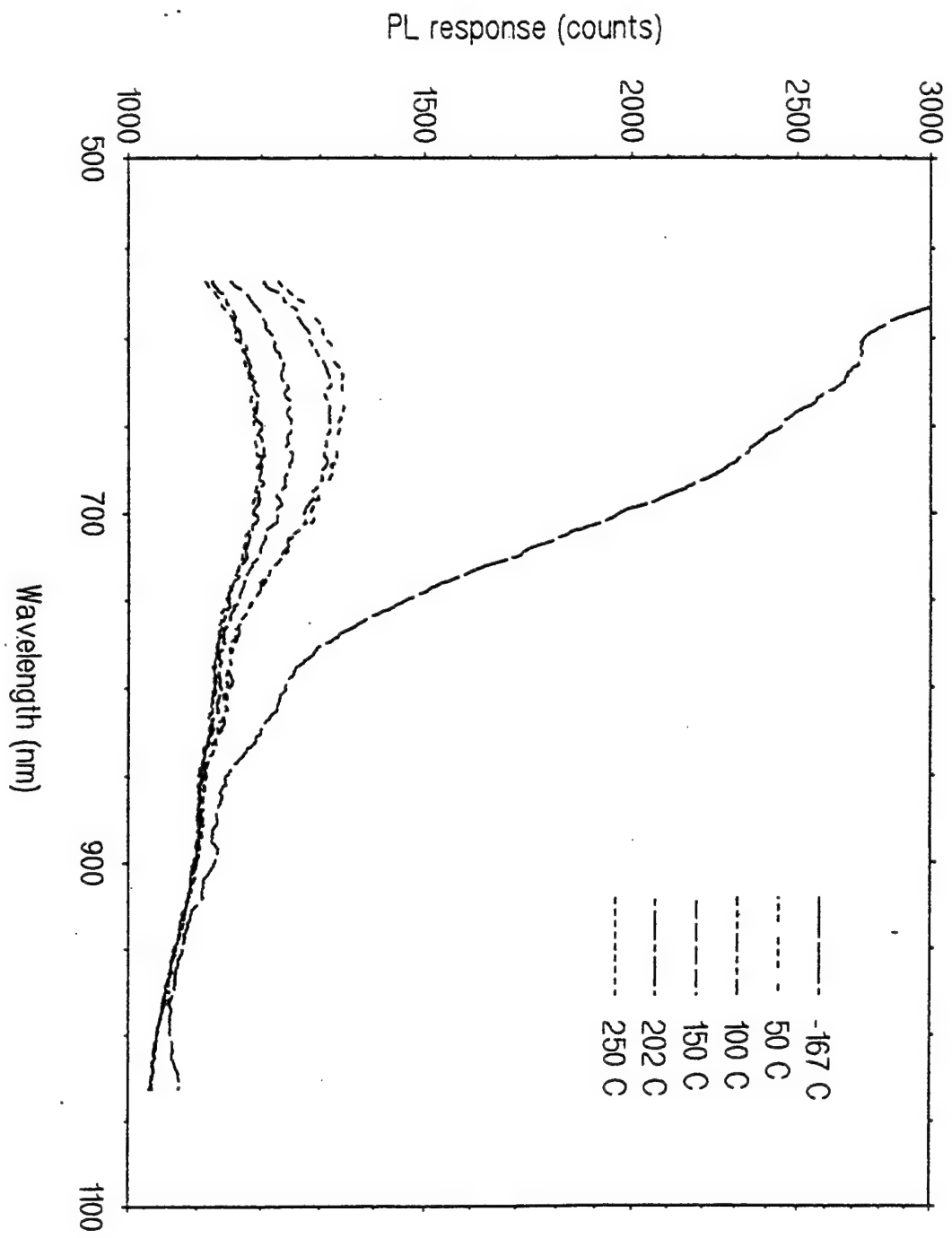
Results and Conclusions

Most of the spectra were well fitted with only three Gaussians, but others, namely those below 4° C, required four. It appeared, therefore, that there was a fourth, separate, light-emitting process that occurred only at lower

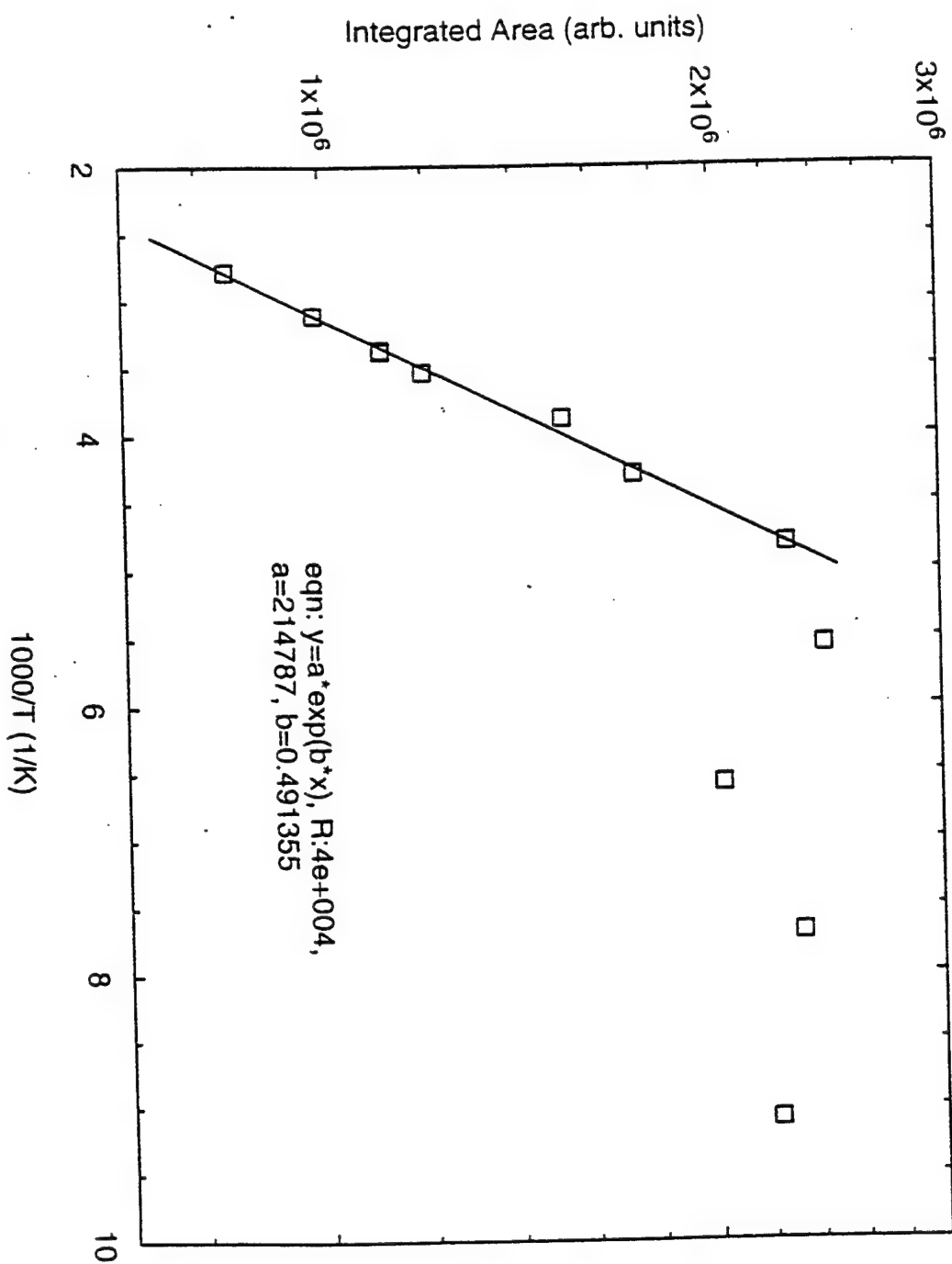
temperatures. As for the Easy Plot graphs described above, they looked very orderly, but not quite explainable. Some few fit well with Arrhenius equations, others to exponential equations, still others matched half-parabolas; many were suited best with simple linear equations; but most fit no equation at all. In addition, we still had no answer as to what was causing the anomalies, so we could not tell what to change to improve our spectra. Still, versus increasing temperature, the energy of each distinct peak would shift upwards, as is usually the case--when the heat energy of a system is increased, more of it is bound to be translated into other forms of energy. The width tended to first decrease slightly then increase slightly, while the intensity remained steady at first then fell drastically as temperature increased; thus the curve of the integrated area followed that of the intensity, as the effect of the width was overwhelmed. The result then, was that as temperature decreased the luminescence was brighter but the color was less blue.

Meanwhile, the dramatic improvement (in relation to that obtained by Samson Jenekhe) in the spectrum of the washed PBT with which these experiments had been done is not completely understood; but more samples have been put down to wash and confirmation is nearing. There is good argument that it is the washing that is responsible for the difference; the ethanol could be rinsing out aggregates and impurities incorporated in the polymer during synthesis or subsequent handling and exposure to air. These irregularities in the polymer could be leading to quenching that distorts and minimizes the luminescence spectrum; in that case, their removal would have the effect discovered in the experiment.

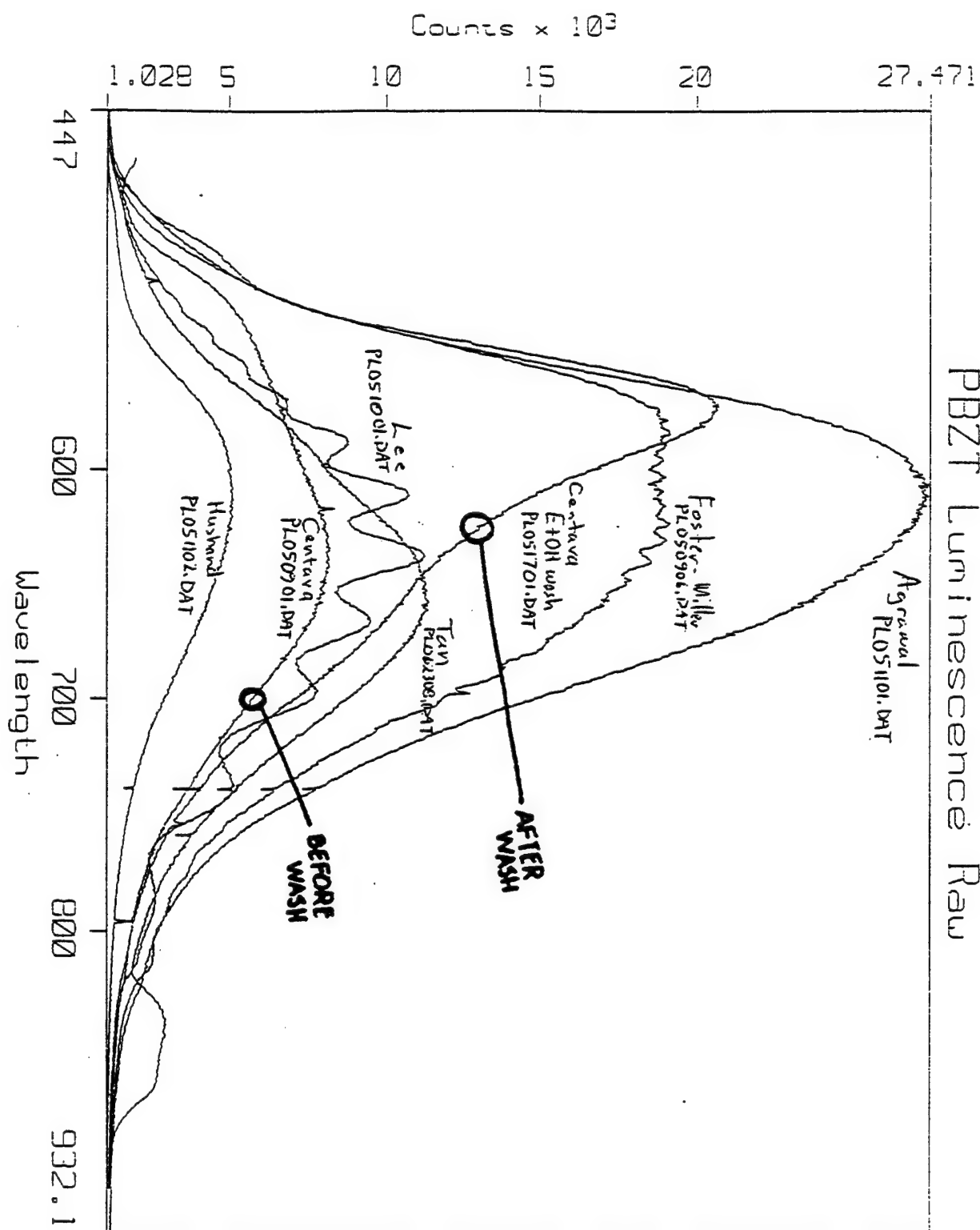
PL of PBZT as a function of Temperature



Integrated Area vs 1000/T for Bruce Co-Polymer



PBZT Luminescence Raw



WORKS CONSULTED

Shigeo Tazuke, Mitchell A. Winnik. "Flourescence and Phosphorescence Spectroscopy in Polymer Systems: A General Introduction," Photophysical and Photochemical Tools in Polymer Science. D. Reidel Publishing Co., 1986. 15-42.

A PROGRAM FOR CREATING BIBLIOGRAPHIES

Ann Nee

Centerville High School
500 E. Franklin Street
Centerville, OH 45459

Final Report for:
High School Apprentice Program
Wright Laboratories

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratories

August 1995

A PROGRAM FOR CREATING BIBLIOGRAPHIES

Ann Nee
Centerville High School

Abstract

A program for compiling and referencing bibliography information was written in Visual Basic. The program consisted of an entry and a browse routine, as well as a routine to create new databases. Its main purpose was to decrease the time that must be spent entering and retrieving information.

A PROGRAM FOR CREATING BIBLIOGRAPHIES

Ann Nee

Introduction

Although doubtless bibliography programs exist, it is almost always easier to use one that has been made specifically for the purpose to which it will be put. This program, then, although it could very well have more diversified uses, was designed for a large collection of scientific articles whose names, authors and publishing information would likely be needed for reference.

Features and Explanations

SETNEW









Enter new database name:

OK

Cancel

Since I was not using the professional version of Visual Basic (which can create new databases on demand), one database had to be set up that had the necessary form, and could be recopied each time a user wished to begin a fresh bibliography. The user had only to type in the name of his choice (excepting only those already existing and the key prototype to be copied), and Setnew would affix the proper three letter description for a database (mdb), and create a new file in the directory in which the application was running.

ENTRY

Book/Journal	Primary Author		List1
Secondary Author(s)			
Article/Chapter Title			
Journal/Book Title		Volume/Edition	
Editor(s)		Total records	
Publisher, City	Page(s)	Year	
  			
Print Commands			
  			

The second program, called Entry, was, obviously, one for entering information into the database's predetermined structure. After a user opened the application, he would be presented with an Open common dialog box where he could choose the database he wished to edit. If he entered a valid existing database, Entry would display the basic form, including a text box with a preset maximum of characters for each of the following items:

Label	Char.	Label	Char.
Type-J(ournal)/B(ook)	1	Volume, Number	8
Primary Author	30	Editor(s)	45
Secondary Author(s)	80	Publisher, City	35
Article/Chapter Title	120	Page(s)	9
Journal/Book Title	80	Year	4

Any characters entered that exceed these parameters are simply ignored. The last record in a database (or the first in the case of a new database) has the pound (#) symbol entered in each box, giving Entry an easy way to recognize when it is at the end of a record set. Without this safeguard, Entry has a tendency to run amok: sometimes overwriting previous entries, sometimes storing doubles of the latest screen, and always miscounting the number of records in the database. New entries are typed over the # symbols in each box; and, when the Update button is clicked, new entries or changes made in an entry are automatically saved. In the former, the record count advances by one and the screen of #s is refreshed.

The final program was designed to allow searches for keywords in the bibliography, and was dubbed Query. At present, Query has been fed the code to search the primary author, secondary author, and article title lines for the word typed in the search string. When Search for Next is visible, clicking on it will bring up the next entry corresponding to the search string, until there are no more.

Items in common between Entry and Query are the record tally and the print functions. A user may choose to print either the entire database or just the current screen by clicking one of the first two buttons in the frame. If the entire bibliography is chosen, then the AllEntries subroutine will first

QUERY

Book/Journal	Primary Author	<div style="border: 1px solid black; padding: 5px; width: 80px; height: 50px; margin: 0 auto;">List1</div> <div style="border: 1px solid black; width: 30px; height: 30px; margin: 5px auto; text-align: center; line-height: 30px;"> </div> <div style="border: 1px solid black; width: 80px; height: 40px; margin: 5px auto; text-align: center; line-height: 40px;"> </div>
Secondary Author(s)		
Article/Chapter Title		
Journal/Book Title	Volume/Edition	<div style="border: 1px solid black; padding: 5px; width: 80px; height: 40px; margin: 0 auto; text-align: center; line-height: 40px;"> </div>
Editor(s)		
Publisher, City	Page(s)	
		Year
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="display: flex; align-items: center;"> Data </div> <div style="display: flex; gap: 10px;"> <div style="border: 1px solid black; padding: 2px 5px; text-align: center;">Search on Author</div> <div style="border: 1px solid black; padding: 2px 5px; text-align: center;">Search on Title</div> <div style="border: 1px solid black; padding: 2px 5px; text-align: center;">Search on Year</div> </div> </div> <div style="margin-top: 5px;"> <input checked="" type="radio"/> Primary <input type="radio"/> All Authors </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px; text-align: center;"> <div style="background-color: #333; color: white; padding: 2px 10px; display: inline-block;">Search String</div> </div>		
<div style="border: 1px solid black; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> Print Commands <div style="display: flex; gap: 10px;"> <div style="border: 1px solid black; padding: 2px 5px; text-align: center;">All Entries</div> <div style="border: 1px solid black; padding: 2px 5px; text-align: center;">Current</div> <div style="border: 1px solid black; padding: 2px 5px; text-align: center;">Print</div> </div> </div> </div>		<div style="border: 1px solid black; padding: 5px; width: 80px; height: 30px; margin: 0 auto; text-align: center; line-height: 30px;"> </div>

alphabetize the entries by the last name of the primary author (if there are multiple articles by the same author, AllEntries will not differentiate, and will return that particular portion in the order it was entered), then print to the printer:

[For J] Primary Author, Secondary Author. Article Title, Journal Title. Volume. Page. Year.

[For B] Primary Author, Secondary Author. Chapter Title, Book Title. Editor. Publisher, Year. Page.

If the user wants only a few select records, he may search--and press the Current button--for each one. When finished, he need only click on Print, and

all his selections will print at once. If either AllEntries or Current is chosen at any time while the program is running, without a subsequent Print, the printer will disgorge all unprinted material upon exiting. The ability to search for spaces was integral to both the alphabetization and the printing. Since proper names are entered in normal order and might or might not have initials which might or might not have periods or spaces between them (dependent on the user), the computer was taught to look for the first space from the end of the primary author's name, after trimming any excess spaces on either side. Reversing the order then became an easy matter; and, after alphabetization by last name, the computer reverted the name to original order and matched it with the appropriate record. In printing, the computer had to recognize spaces in order to execute effective word wrapping. After each space in the entry, Print would test the length of the text to determine if the entire string would fit on one line. If not, Print would move the last word to the next line and start again. Left, top, and right margins were set at roughly one inch, and the bottom margin was allowed to fluctuate (to cut down on entries being sliced in half by a new page) with an average of about an inch, as well.

Conclusion

There are a few convenient advantages of this program over typing a bibliography in a word processing program. For one, all the spacing, punctuation, and order are already taken care of; there is no need to worry about whether a certain piece of information is probably required, or where to put it. For another, it is no longer necessary to enter your works in alphabetical order no sorting through the whole pile beforehand, and no need to wear out the up and down arrows, organizing as you type. And again, the

program will set the lower margin to maintain the fewest divided entries, to improve reading ease and prevent one-printed-line pages. And yet another, finding an entry, or all entries by the same author, becomes a matter of typing a handful of words instead of rereading the entire list. The true usefulness of this program becomes most evident when dealing with large amounts of material.

END

A STUDY OF THE FLIGHT DYNAMICS REFERENCE SYSTEM
FOR WRIGHT LABORATORY

David M. Pepin

Greenon High School
3950 South Tecumseh Road
Springfield, OH 45502

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

A STUDY OF THE FLIGHT DYNAMICS REFERENCE SYSTEM FOR WRIGHT LABORATORY

David M. Pepin
Greenon High School

Abstract

The reference system made by the Flight Dynamics division of Wright Laboratory was studied and improved. The system was put together several years ago using DBase IV, for the purpose of making it easier to find previously written reports. Over the course of several years, the reference system began to lose popularity, and was nearly forgotten altogether. It was decided that in order to get it started again, it needed to be transferred to Microsoft Access so that it could be used in Microsoft Windows. The ultimate goals were 1) to successfully transfer the database from DBase IV to Microsoft Access, 2) to enter several recent reports into the system, and 3) to get people using the system again to find reports and enter some of their own.

A STUDY OF THE FLIGHT DYNAMICS REFERENCE SYSTEM FOR WRIGHT LABORATORY

David M. Pepin

Several years ago, a reference system was developed for the FIGD branch of Wright Laboratory. The system was created in DBASE IV and had many useful functions. It was designed in such a way that anyone in the building could use a computer to find a document that they were interested in, and then find that document on a shelf. This was useful for anyone who was doing research in a particular field, because they could simply look to see if anyone had done similar research before.

The fields in the main table included title, author, date, product number, sequence number, location, keywords, and comments. The title, author, and date were the primary data referring to the document. The comments were included if the author thought that it would help explain the content of the report. The keywords, which will be discussed later in more detail, were used only for picking the specific document from a list. The product number, reference number, and location were used to determine the document's type, and where it could be found on the book shelves.

There were several features of this reference system that made it user-friendly. The main option available was a search, in which the user could input several parameters, and the computer would list all matches found. Looking through the entire table row by row to find a specific document was not acceptable. There were three ways in which the user could search for a report. The first way was to search by author. The computer would prompt the user to type in a name, and would then display all records with that name listed in the author field. Another way to search for a report was by the title of the report. If you knew the exact title of the document you were looking for, this was ideal because you would never get more than one match. This is because no two documents ever had exactly the same title. The third and final way to pick out certain reports was to match keywords. Each report had anywhere from one to five keywords associated with it. This way of isolating reports was different from the rest in one very important way. When the user chose a keyword to match, he did so by choosing a keyword from a list. There were over twenty different categories of keywords, and many keywords within each of those. The advantage of choosing the keywords in this manner was that the user would always find a match. When the data is typed in by

hand, there is always the chance of misspelling it, or simply choosing a keyword that doesn't exist. Among these three ways of narrowing down the documents, it usually made it easier for the user to find the specific one of interest to him.

There was one person in charge of the reference system known as the administrator. The database included several options available to this administrator. These options included adding or deleting a report from the main table, adding a keyword to a keyword category, and adding a new keyword category. The most common of these was adding a document to the reference system. This was done in two steps. First, the person who had the document had to fill out a form, in which they listed the necessary data about the report. They would then give this form to the reference system administrator. The administrator would then input this data to the computer. This took care of all but two of the data fields.

One of these exceptions was the sequence number. Each report had a product number, which was a combination of six letters. The first two of these, known as the product type, indicated the kind of document it was. For example, the letters TR meant that the document was a technical report. The last four letters, known as the product category, represented the main topic of the document. For instance, if the last four letters were SFTW, it meant that report was about software. There were several different codes for the first two letters as well as the last four. These codes were all listed in two menus, so the user would simply determine what codes to use for their report based on the available choices. Obviously, there would often times be duplicate code combinations. For example, eventually there was going to be more than one technical report written about software. Because of this, the sequence number was developed. The first technical report about software would be given the sequence number one, the second would be two, etc. This way all documents would always have different identification codes. The nice thing about the DBASE IV was that once the user input the product number, the computer would automatically determine the sequence number. This cut down on the possibility of human error.

Another field, the location, was determined by the administrator alone. This number indicated where the document could be found on the reference system shelves. The purpose of this field was to help the user locate a report on the shelves after finding it on the computer. A location could be something like RF-3. The shelves were

labeled and organized in such a way that anyone could find a report based on its location and title.

A second option available to the administrator was removing a record from the reference system. This was usually done if the information in a report was outdated or just not useful anymore. All the administrator had to do was type in the title of the document when prompted to do so. The database automatically updated any and all sequence numbers as necessary.

The final option was adding and deleting keywords and keyword categories. Occasionally, it became necessary to add a keyword or an entire keyword category. This happened in some cases when something new was being studied or developed, and there were no keywords available that would adequately describe the report. Sometimes if the new keyword didn't seem to fit into any of the existing keyword categories, it would be necessary to add a completely new one. Deleting keywords and keyword categories was possible, but not nearly as easy as adding them. Many complicated steps were involved which made it a most undesirable thing to do. In most cases, because of this, once a keyword or keyword category was added, it would usually stay there indefinitely.

For some time, the reference system was serving its purpose quite well. There was a special room in which all the records were kept, and new records were added to the system regularly. Unfortunately, the person who developed and ran the reference system was transferred to another location. Soon after this, people began to lose interest in the system. For a while, whenever they had a report that could be put in the system, they would just put it back in the room where all the others were. Other times, they would go back to the room, find a useful report, take it out, and never return it. This soon caused a large and complicated mess. Eventually, almost everyone forgot about the system altogether, and simply kept all important documents in their own offices.

This is where I came into the picture. It was decided that the reference system could be revived if it were put on Microsoft Access. The reason for choosing a database like Microsoft Access was that it could run in Microsoft Windows, and that anyone could use it through a shared directory on a network. No one else really had the time to do this, so it seemed the perfect job for a high school apprentice. The person who originally created the system was now back in the building again, so I would have help figuring out the old system if necessary.

Microsoft Access is much different from DBASE IV. There are six main components to a full database in Microsoft Access: tables, queries, forms, reports, macros, and modules. Of those, I used one table, five queries, two forms, and three macros to make the reference system. A table is a spread sheet with fields and records in the rows and columns. A query is used to make a temporary table, using only the records that match certain criteria. A form is used to allow the user to choose from several things, or to input data in an organized fashion. Reports are used mostly for documentation and special comments from the specific database. Macros are used as pointers in the database. In other words, a macro can tell the database to carry out commands, such as opening a table or running a query. I do not know exactly what modules do, because I didn't find it necessary to use any of them in the database I created.

My main goal was to make the reference system in Microsoft Access as useful and as user-friendly as it was in DBASE IV. Some of the files and features of the original system could simply be converted to a Microsoft Access format, but others had to be done from scratch. The first task was to get the main table from DBASE IV to Microsoft Access. The table itself is the heart of the entire reference system. This was accomplished by importing the table into a new database in Microsoft Access. Microsoft Access was able to automatically convert the table from the DBASE IV format. Unfortunately, this was the only part of the old reference system that could be easily converted

After this, the rest of the improvements made to the new database were all made for the purpose of making the system more user-friendly. The first goal was to set up menus like those in the old database, in order to make it easier for the user to operate. This was accomplished through the use of forms in Microsoft Access. I was able to create two forms that give the user and administrator the same options as the previous database did. The first form gives the user four choices: search for a report, add a report, delete a report, and check out/return a report. The final option, check out/return a report was not part of the original database. It was hoped that this option would make it easier to keep track of the reports in the library. The second form menu gives the user the choice of searching for a report by title, author, or keywords, or listing all of the reports. I included buttons with these forms so that all the

user may simply point and click to choose any one of them.

The next step was to tell the database what to do when the user chose any one of the options. The most difficult of these were the search options. The database had to know how to search for reports in all of the different ways listed previously. This was when the queries and macros became necessary. I made one query for each of the search methods, and used macros to define the connection between pressing a button and running the appropriate query. The search by title query prompts the user to type the title of the report that they were looking for, and then displays all fields pertaining to the report with that title (if a match was made). The search by author does the same thing by prompting for the name of the author. The only difference is that the database may find more than one match. These two options are almost identical to their counterparts from the old system. The search by keywords is different in that the user is prompted for up to five keywords.

There are two restrictions on the search by keywords query that didn't exist on the old database. The first of these is that the user must type in the keywords, instead of choosing them from a list. This problem is mostly eliminated by a printed list of all keywords setting beside the computer. The second limitation is that Microsoft Access can only do an 'or' search when using multiple fields. In other words, if the user was to type in three keywords, the database would find all reports that had any of the three associated with it. An 'and' search was not possible because the query would automatically try to match the user's **first** keyword with the report's **first** keyword, the user's **second** keyword with the report's **second** keyword, etc. This is simply the way queries are set up in Microsoft Access. I believe there may be a way to overcome this problem through the use of SQL (a sort of programming language used in queries), but I was not able to discover it. After the search parameters are typed in by the user, a temporary table containing all records matching the user's input will open. Besides the two limitations on the search by keywords query, the search options acted much like they did in the old database.

Now that the searches were taken care of for the most part, the next step was to make it possible to check out and return a report from the reference system, in hope of making the system act more like a library. The first step was to add two new fields to the main table; 'Checked Out By' and 'Date Out.' When the user chooses the Check

Out/Return option from the main menu, the database will run the corresponding macro. This macro then runs the appropriate query, which opens the main table and allows the user to edit these two fields. When the user wishes to check out a report, he can type his name and the current date into these fields, and if he wishes to return a report that he checked out earlier, he can delete the information from those two fields. Obviously this process relies heavily on the honor system, but that won't likely be a problem.

The two previously listed options, Search for a Report and Check Out/Return a Report, are available to all users. The other two options, Add a Report and Delete a Report, are used by the administrator only. A way had to be developed in which the options available to the user were determined by the access level of the user. This problem was solved by means of a logon dialog box. Microsoft Access gives the option of adding users. What this means is that upon entry of Microsoft Access, a window will appear prompting the user for his name and password. I made one user named Guest, that all normal users can login as. There is no password associated with Guest, so all the user has to do is type 'Guest' in the name field and continue. I made a second user named Admin, which does have a password. Admin is for use of the administrator only. There is also a way of assigning specific privileges to different users in Microsoft Access. Guest has permission only to search for, check out, and return reports. Admin, on the other hand, can use all options and can even change the setup of the entire database.

The next task at hand was to set up ways for the administrator to add and delete reports. At first, the best way to do this seemed to be through the use of forms and queries, but there were some unanticipated problems. For example, in order for a form to change the data in a table, it must be declared at the beginning that the form is based on that table. The recurring problem was that any time I tried to base a form on a table, I always got a 'General Protection Fault' and had to exit Microsoft Access completely. I believe this problem was caused by the fact that there were only four megabytes of memory on my computer. Unfortunately, I never got around this problem. There was an alternative, but it wasn't at all desirable. If the administrator wants to add or remove a report from the reference system by choosing the appropriate option from the main menu, the database will simply open the main table and let him edit it in any way necessary. This means that any time the administrator needs to add a document to the system, he must type all of the necessary data directly into the main table. This isn't nearly as user-

friendly as the old system, but it does still gets the job done.

The computer part of the database was now nearly complete. The remaining problem was the organization of the documents themselves. The room where they had been kept before was a complete mess. There are eight large file cabinets in this room, most of which were full of documents and reports. The problem was, some were part of the reference system, but most of them still needed to be entered into the system. From the last time the system was updated, the reference system contained 312 documents. The next job was to inventory all the documents in the room, determine which were part of the reference system and which were not, and to figure out which documents were missing. Eventually, it was discovered that less than half of the reports in the reference system were present in this room, which meant that most of the others were dispersed throughout the building. A second discovery was that there were several hundred documents that needed to be added to the reference system. Another problem was that even the documents that were part of the reference system and were in that room were in no order whatsoever.

The next goal before me was to organize all of the documents in that room. It was decided that it would be best to use a slightly different organization system than the last time. This was quite time consuming, but was eventually accomplished. Meanwhile, many of the missing documents were found in various places around the building and returned to the main room. I did not have time to enter the documents into the library system, so the last job I had to do was explain the new reference system to everyone else that was interested. During the meeting in which I explained how to use the system, someone volunteered to take on the role of administrator. This meant that the new system could now survive without me, and that my job was now done.

All together, the new reference system is not as user-friendly as the old one, but it is still an improvement because it is in Microsoft Access. This means that there are many features that can be improved if anyone is willing to spend the time on it, and that it can easily be upgraded to new versions of Microsoft Access. The ultimate goal is that this system will last longer than the old one. It is hoped that in the not too distant future, the other documents will be added to the system, and people will continue to use the system and add their reports to it. If these things are done, the new reference system should turn out to be an invaluable tool.

A INTRICATE STUDY OF LASERS AND AN
OPTICAL MICROPHONE

Neill W. Perry
WL/MNGS

Crestview High School
1304 N Ferdon Blvd
Crestview, FL 32536

Final Report for:
High School Apprentice Program
Wright Laboratories

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratories

August 1995

AN INTRICATE STUDY OF LASERS AND AN OPTICAL MICROPHONE

Neill W. Perry
Crestview High School

Abstract

Laser light theory and its practical applications were tested. First an in depth study of laser light and its known properties was performed. Experimentation was made easier by gaining a thorough understanding of lasers. The first two experiments were to gain experience with an interferometer by producing interference fringes by combing two coherent laser beams and to measure the coherence length of a Helium - Neon laser, the laser used for this project. Once tests on the interferometer were completed it was modified to become a vibrometer (optical microphone). Audio signals were injected into the optical microphone to see their frequencies. Results proved that the vibrometer could detect the frequency of a 500 Hz signal with ninety -five percent accuracy.

AN INTRICATE STUDY OF LASERS AND AN OPTICAL MICROPHONE

Neill W. Perry

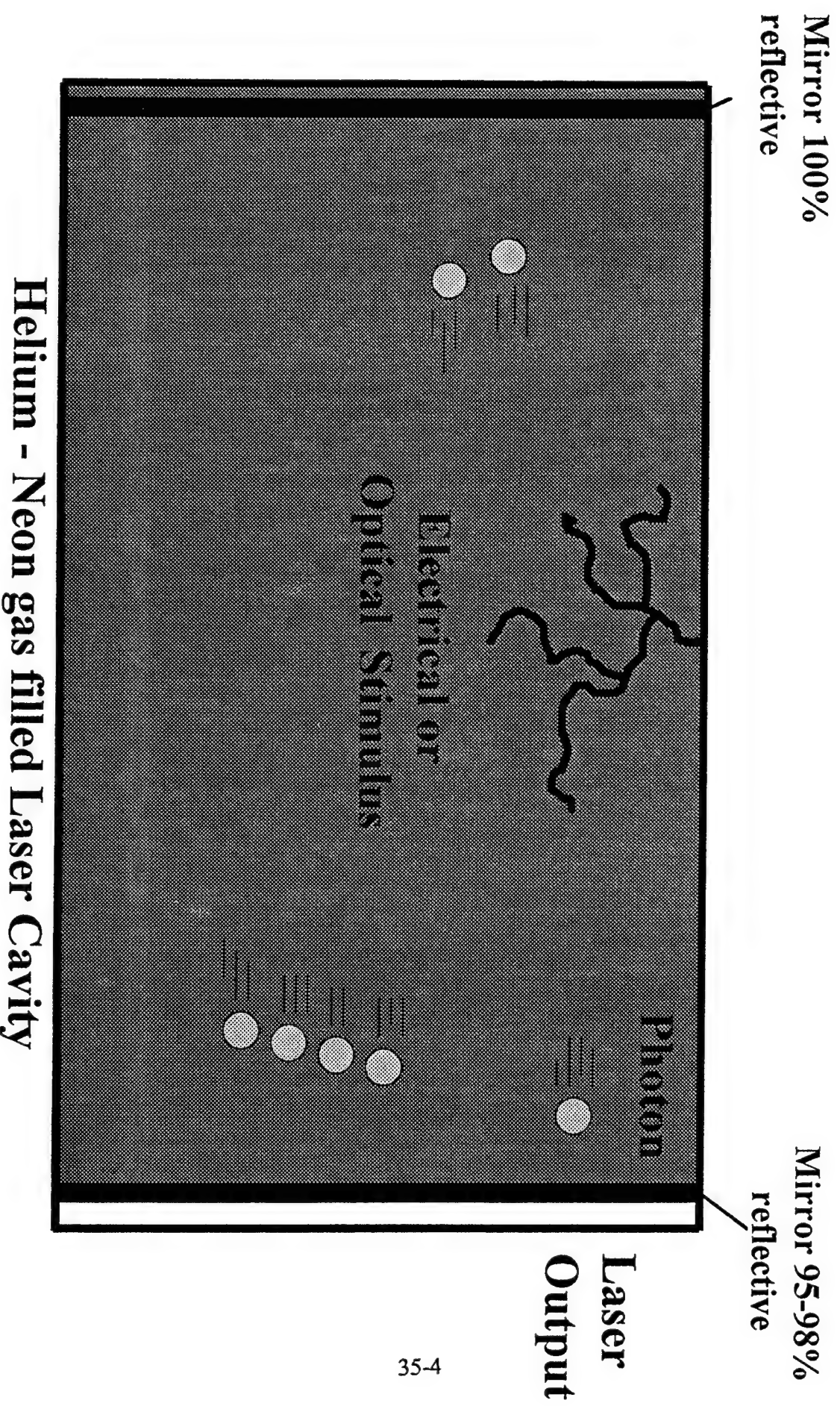
Introduction

Since their invention over thirty years ago lasers have been studied and researched intensely both for theoretical viable and practical applications. Once such application is using a laser in a interferometer. An interferometer should in theory allow detection of interference fringes when two laser beams of the same wavelength are combined and also to measure the wavelength of a given laser. Another practical application is using a laser in a vibrometer, or optical microphone, to detect and measure the frequency of a vibrating object. A vibrometer would enable the user not only to detect such vibrations but also to distinguish between similar audio signals. For example the operator could tell the difference between an automobile with a four cylinder engine and one with a six cylinder engine.

Methodology

The first phase in this project was to research the properties of light which would be studied. Laser is an acronym which means: Light Amplification by the Stimulated Emission of Radiation. Laser light is of all the same wavelength or, monochromatic. Another important property of lasers is over what distance will the light remains in a predictable phase called the coherence length. Laser light is produced from a gas filled cavity (As shown in Figure 1). When the laser is first turned on an electrical or optical

Figure 1



sent throughout the laser cavity which excites the gas atoms. Eventually one of the excited gas atoms emits a photon. The photon travels until it collides with another excited gas atom. As a result of this collision another photon, an exact copy of the original, is produced. These actions sets off a chain reaction of photons cloning themselves exponentially. Within a very short time span the entire laser cavity is filled with an high number of photons in phase with each other and at the same wavelength as the original. A small percentage of these photons are released when they strike a partially reflecting mirror. Photons that pass through are the ones that comprise the laser beam found outside the laser cavity.

Interferometry

The first test conducted was to measure the wavelength of a given laser. The Twyman - Green Interferometer, the one used in the test, is an optical device that separates a laser beam into two beams then rejoins them back together again (as seen in Figure 2). In this device, the beam expander expands the beam as it first exits the laser cavity. Then the beam is split into two independent laser beams by the fifty-fifty beamsplitter. The beams travel their optical pathways and are reflected back over the that same pathway by their respective mirrors. The fifty-fifty beamsplitter recombines the two lasers and directs them toward to the detector.

The theory behind interferometry is that a laser beam has planes of photons in constant phase, called wave fronts (See Figure 3). These photons are all at the same point on the sine wave which describes their propagation trough space and time.

Figure 2

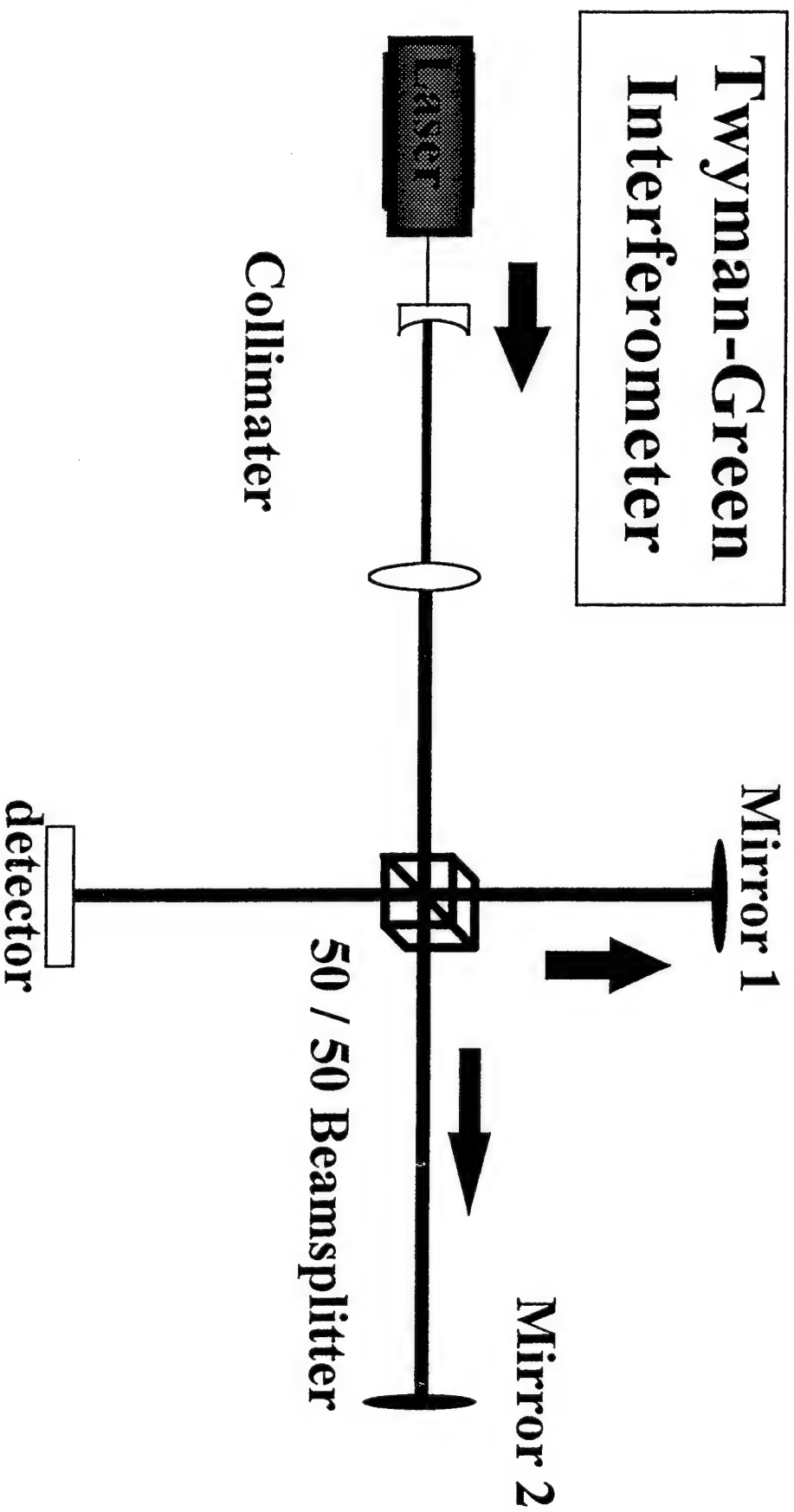
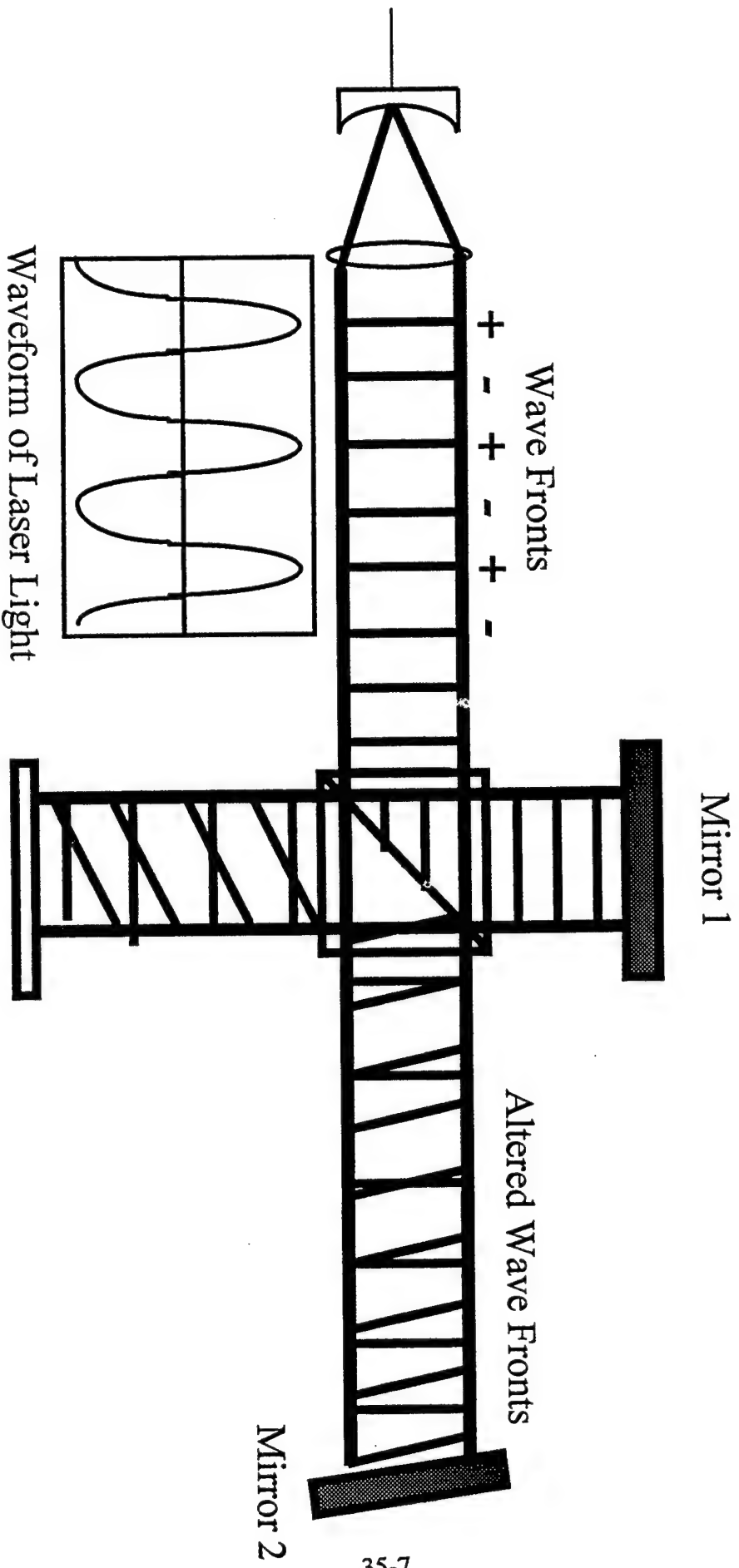


Figure 3



These wave fronts are split at the detector and travel their separate pathways. One beam travels to mirror 1(m1) and is reflected back to the beamsplitter. However the beam that travels toward mirror 2(m2) is reflected back slightly tilted with respect to the first beam. As a result the wave fronts are not completely in alignment when they are combined in the region of interference. These interference patterns which result from the slight tilt, or fringes, can be seen on a reference card or detector(see Figure 5). Areas of darkness are where the two wave fronts are 180 degrees out of phase ; thus the two waves totally cancel each other out or destructively interfere. Points at which the waves are completely in phase constructively interfere and the light is intensified.

Experimentation

The first experiment conducted was to find out how accurately . This was done by decreasing the optical pathlength of one laser beam by 20 microns. As m2 was moved interference fringes that scrolled past a reference point on the detector were counted. By knowing how many waves there were in a given distance the wavelength of the He-Ne laser can be found. The measured wavelength was 606 ± 30 nanometers compared to the established wavelength of 632.8 nanometers.

The second experiment ran on the interferometer was to measure the coherence length of the He-Ne laser. This was done by increasing the optical pathlength of one beam until interference was no longer observed. The coherence length of the laser used was a minimum $9.98 \text{ meters} \pm .05 \text{ m}$. Research indicated this was an acceptable distance for the He-Ne laser.

Figure 4

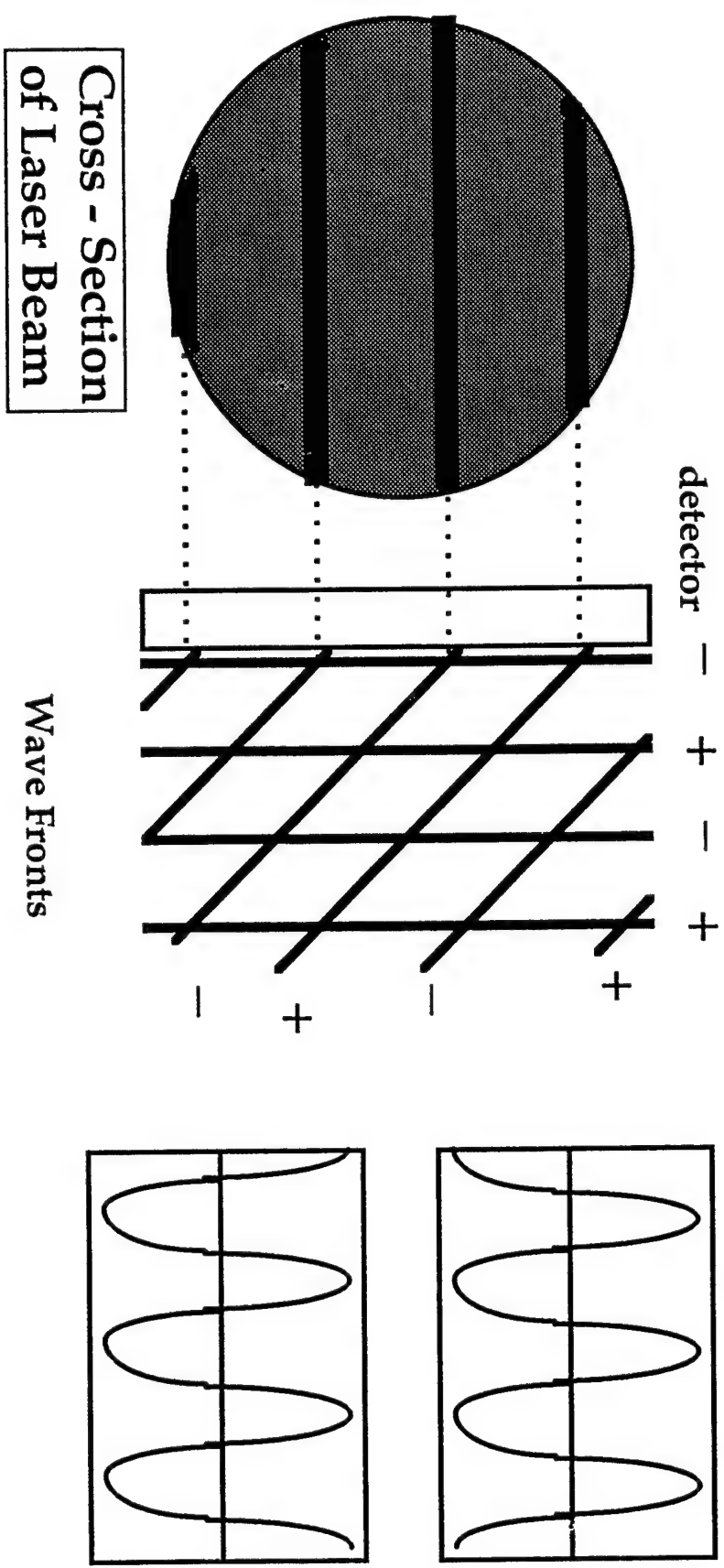
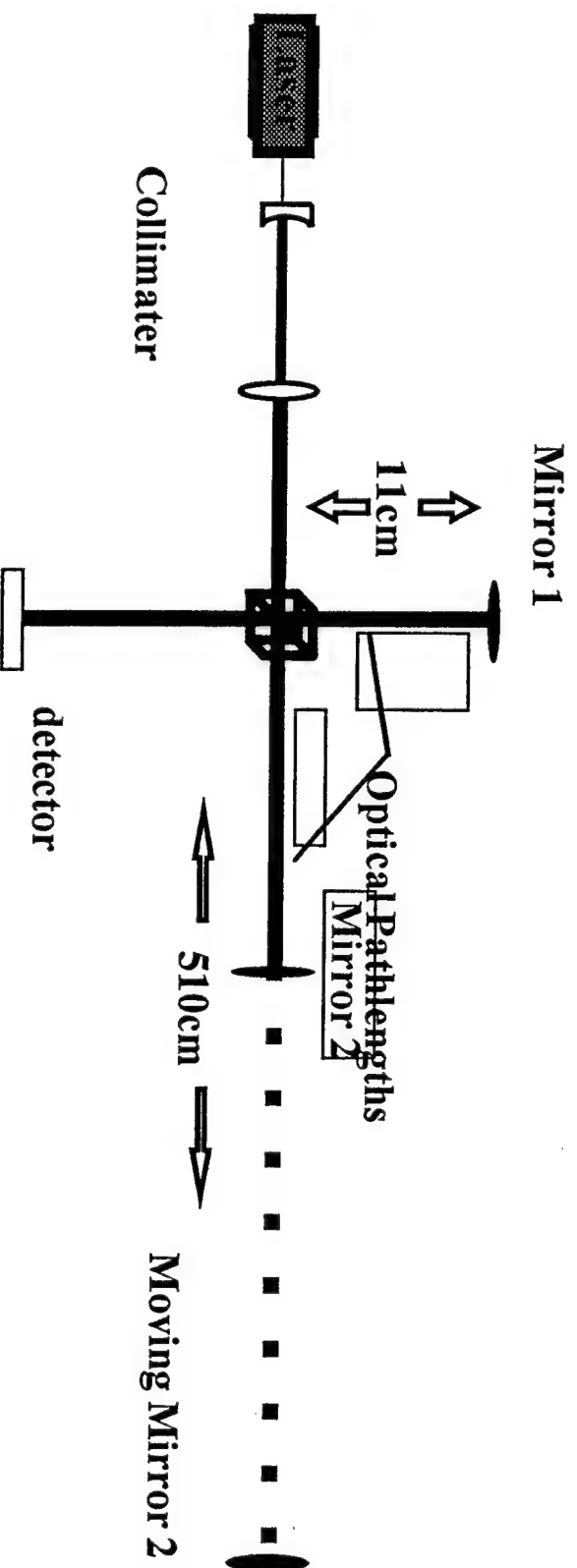


Figure 5

Measuring Coherence Length



$$\text{Coherence length} = \text{Difference of optical pathlengths} = 2(510\text{cm}) - 2(11\text{cm}) = 9.98 \text{ m}$$

Vibrometer

The final stage of this project was to modify a Twyman-Green interferometer into a vibrometer. This was done by adding an audio input into mirror 2, installing a PIN diode detector, and inputting the information from the diode into a digital oscilloscope. By adding an audio signal into the vibrometer m2 vibrates as a result of the sound waves hitting it. This vibration causes the optical pathway to shorten and lengthen by a very small distance; on the order of half a wavelength. The interference fringes move back and forth accordingly. A small portion of light from the detector is sent to the PIN diode. The diode determines the frequency by measuring the intensity of light falling on it, allowing one to distinguish from a light or dark area. A digital oscilloscope displays the measured waveform.

Experimentation

The experiments ran on the optical microphone consisted of inputting audio signals or placing a mirror onto a vibrating object. The oscilloscope would then display the vibrational signature gathered. Figure 8 shows the signature of a musical sample (Elvis Presley's Jailhouse Rock) inputted into the vibrometer. The simplest waveform collected was from the tuning fork (Figure 10). The tuning fork was a C note with a known frequency of 523.3 Hz. From the waveform collected the measured frequency was calculated to be 525 ± 15 Hz, thus demonstrating that the optical microphone's ability to accurately measure the frequency of a given signal. In Figure 10 the waveform of a Volkswagen Jetta's engine being started is shown.

Figure 6

The Vibrometer

A vibrometer is a modified interferometer that has an audio signal injected onto a mirror. This input causes the mirror to vibrate or move back and forth a minute distance. A PIN diode is also installed to designate the difference between light and dark on the fringe pattern.

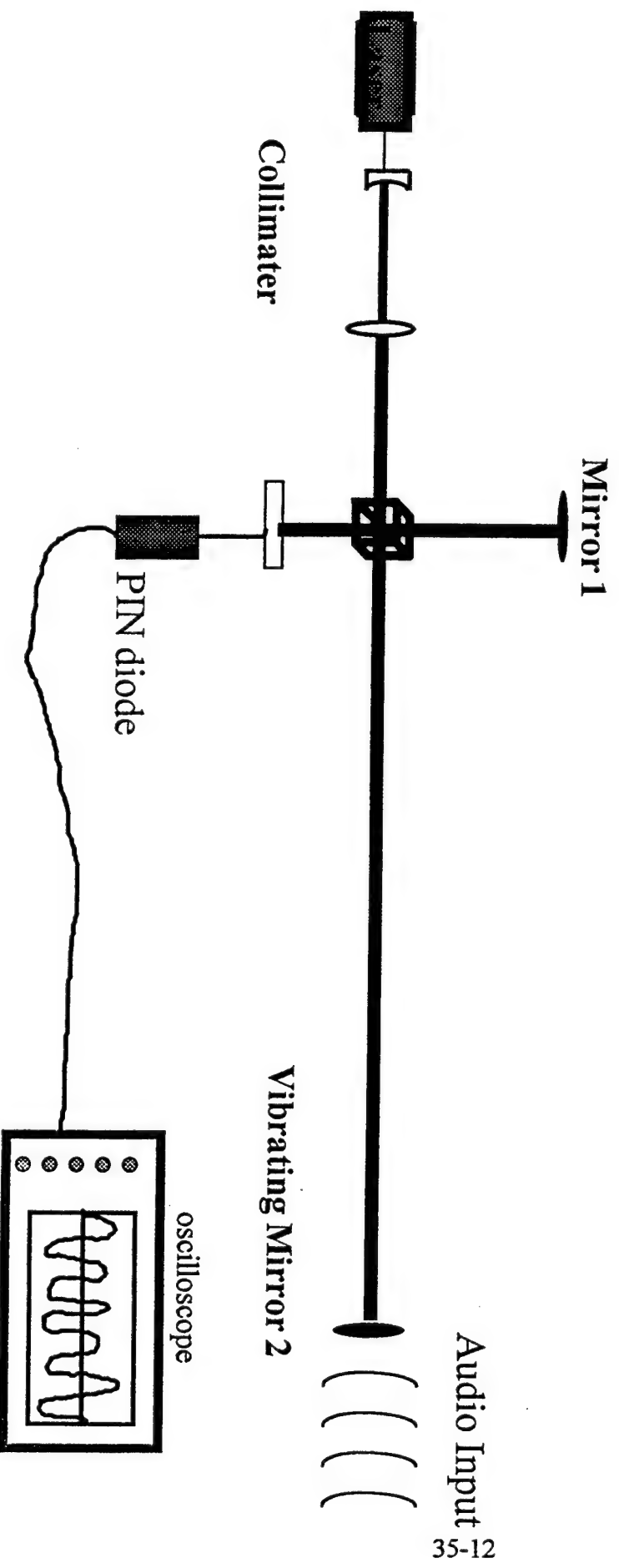
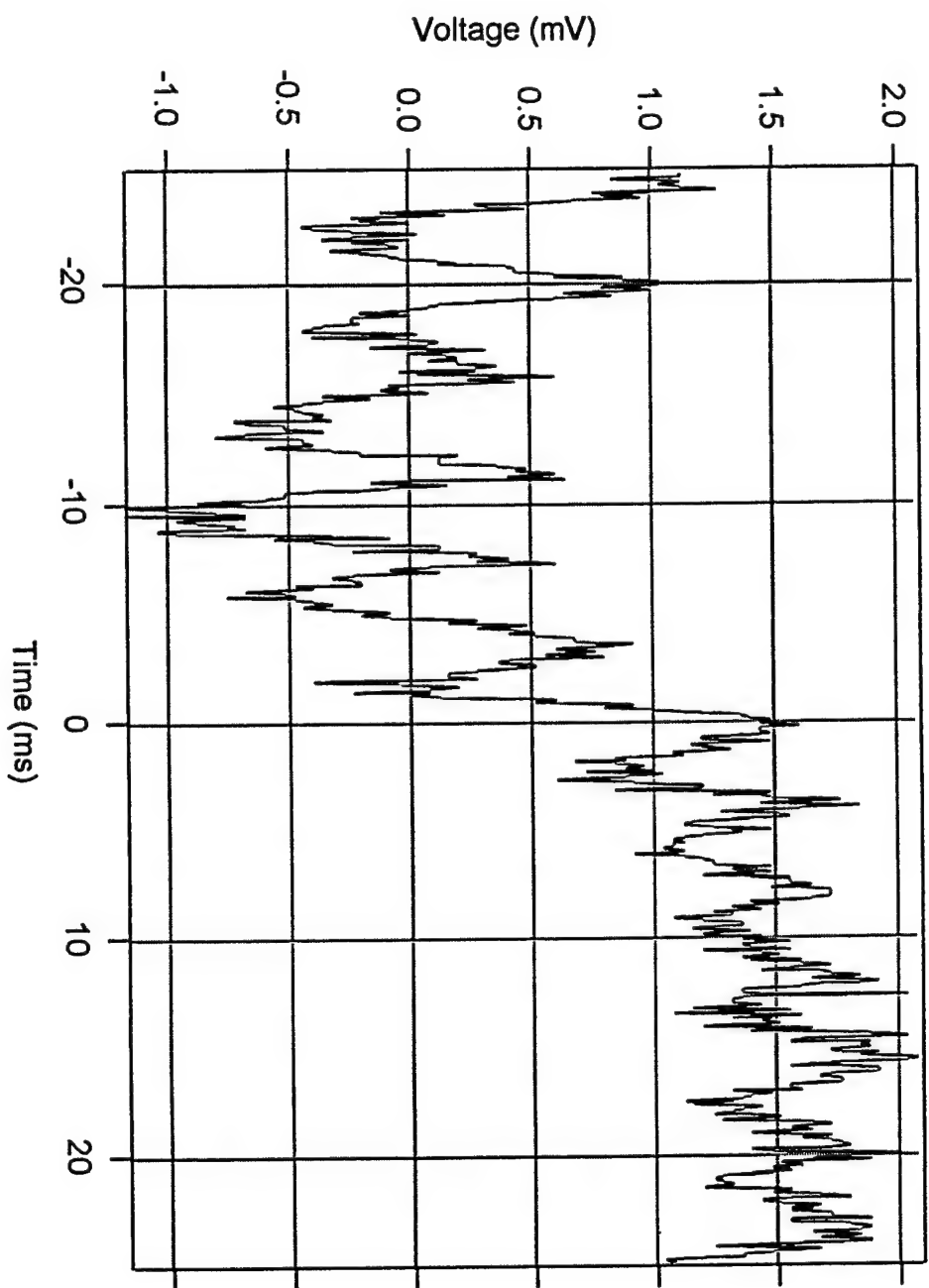
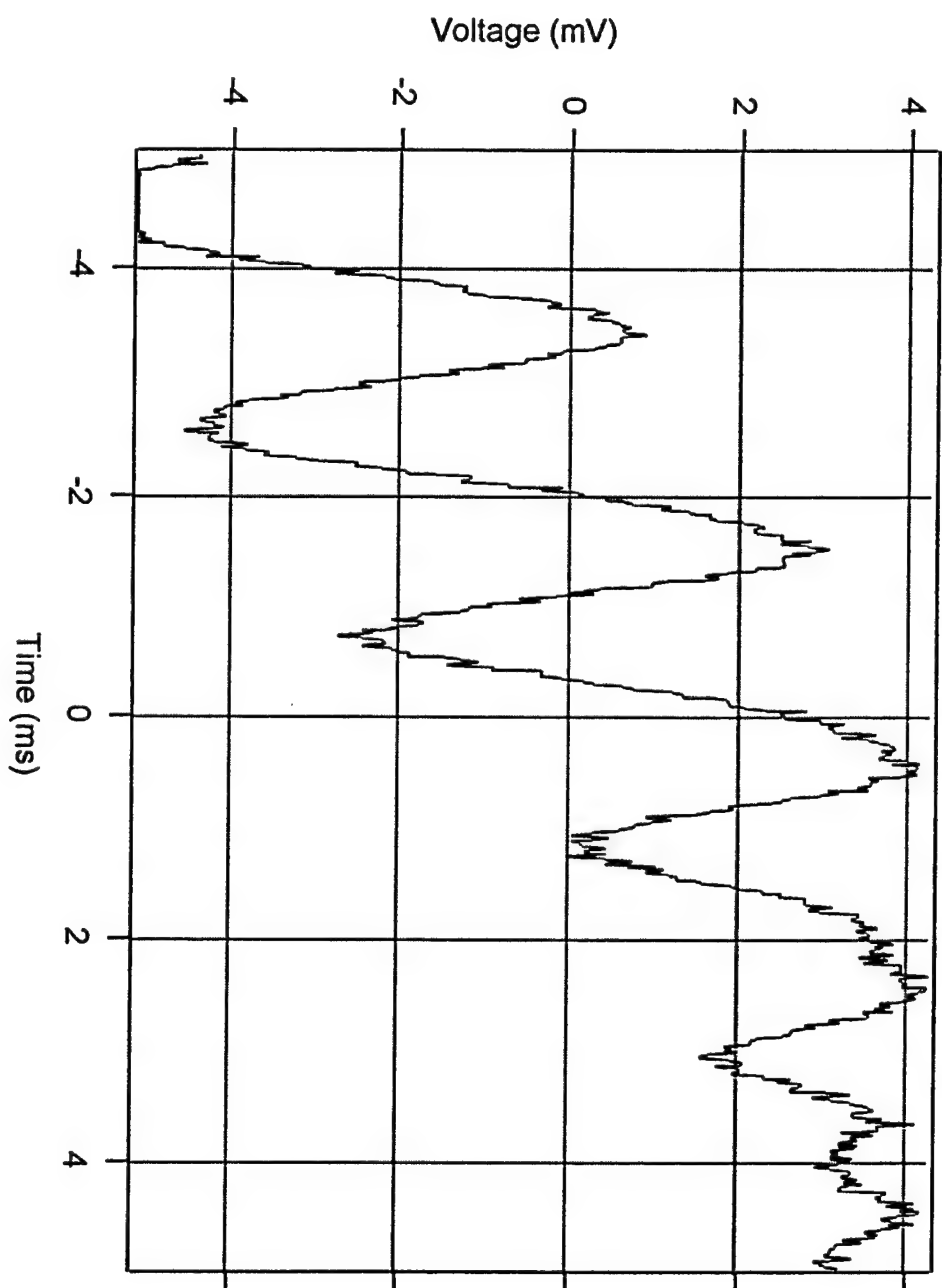


Figure 7



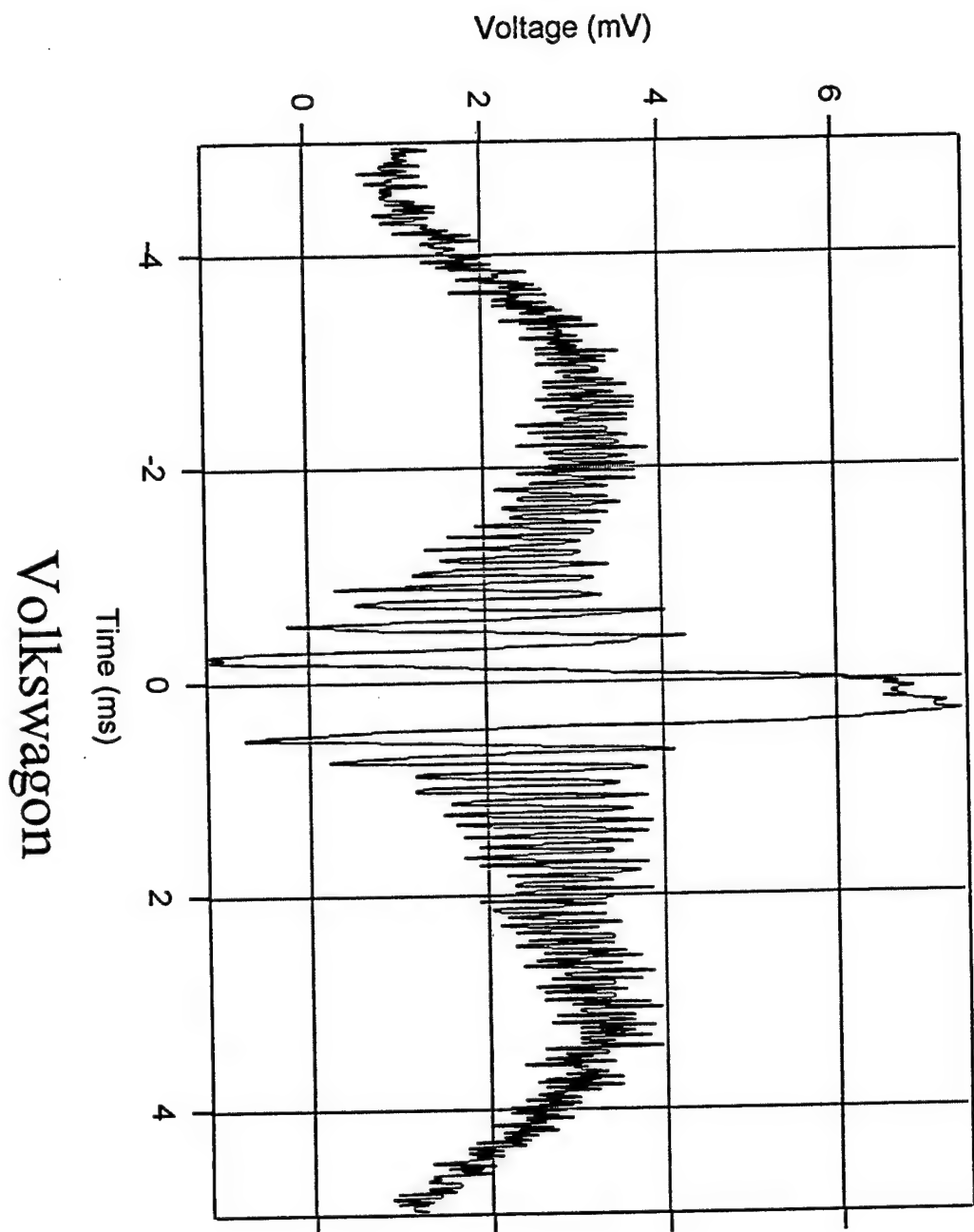
Music Sample 1

Figure 8



Tuning Fork

Figure 9



Conclusions

From the experiments conducted and the data gathered it was shown that an interferometer can successfully detect the presence of interference when two laser beams of the same wavelength are joined and that by using an interferometer the wavelength and coherence length of a given laser can be measured. Furthermore it was proven that an optical microphone, or vibrometer, can measure the frequency of a vibrating object.

Bibliography

Hect, Jeff Understanding Lasers. W. Sams & Company 1988. Radio Shack.

TESTING OF ABTECH'S AIM SOFTWARE

Shaun Power

**Heritage Christian School
325 Ledbetter road
Xenia, Ohio 45385**

**Final Report for:
High School Apprenticeship Program
Wright Laboratory
WL/AARA**

**Sponsored by
Air Force Office of Scientific Research
Bolling Air Force Base, Washington DC**

and

Wright Laboratory

August 1995

TESTING OF ABTECH'S AIM SOFTWARE

Shaun Power
Heritage Christian School

Abstract

The technologies that allow us to build models automatically from exemplary data are important for many computational tasks such as building target recognition algorithms. A leading example of automatic data modeling is Abtech Corporation's Abductive Information Modeler (AIM) (C). AIM builds networks using low polynomial nodes in an architecture that is based on the input data. According to Abtech, AIM should be able to smooth continuous relationships. AIM has been applied to DoD problems in over a dozen funded efforts. However, there has never been any published study that demonstrates AIM's performance on a systematically generated benchmark set of functions. This study was an attempt to fill that gap, this would allow the Air Force (and others) to make better judgment about the applicability of Aim to a given problem. The benchmark functions used in this study were all of the form $(x_1-1)(x_1-2)(x_1-3)(x_2-4)$. We studied the effect of changing learning parameters, number of training samples, range of input, variable values, and benchmark functions. In summary, we found that the CPM changes had no effect to our test, and the number of training samples had little effect except when an odd combination of polynomial and range. Surprisingly, the input range was a very important factor for higher polynomials. We also found that AIM modeled the benchmark functions very well up to an equation with an 8th variable up to the 3rd order. In conclusion, we recommend a number of open problems for testing which were identified during this study.

Introduction

The AIM software by Abtech is an abductive reasoning polynomial net software (see reference 1). This particular version takes polynomial data and generates and synthesizes its own training data. From that point, it can create a learned function or solution to the original input. In testing, we wanted to find a point where the software could not calculate the solution for the input correctly.

General Experiment Design

To test the software effectively, we needed a specific method to begin our testing:

1. Creating the raw data.

This was done by using a simple C program or a Microsoft Excel spread sheet. The C program was used on the smaller equations, until the program became so complex that writing the program could create an error.

2. Training AIM.

The training of AIM consisted of three things; importing the data, generating the data and synthesizing the data. In these three steps, Aim takes the data and trains itself on the input. It then creates its own data, and finds the best solution for that input. The evaluation AIM provides is a category of statistics on the learned solution that AIM produces.

3. Analysis.

In this step, we needed to determine if AIM was producing accurate solutions. In the first couple of analyses we figured out if the function that Aim was producing was similar to the original input by

solving the equation AIM provides. This process helped us determine the size of the error statistics AIM was producing. After an error size was determined, we compared different tests to see what direction AIM was heading.

Impact of different variables

In order to see how to test AIM effectively, we ran a few simple polynomial tests. In those first tests, we came across some problems that helped manipulate changes for our further testing. The solutions to those problems became our defaults, or the given changes we had to make in order to prepare each set of data for testing.

Rows

Problem: What do changes in the number of rows do to AIM's results ?

Variables for the 1st test:

Data Generation	C program
Polynomial	$y = c0 + c1 * x1 + c2 * x1^2 + c3 * x1^3$
Coefficients	$c0 = 1$
	$c1 = 9$
	$c2 = 7$
	$c3 = 0$
Number of Rows	102
Input Range	1000 to -1000
CPM	1
Root location	1100 to -1100
Analysis	Based results on Average Absolute Error

Results: Average Absolute Error = 4.1094 (Low Error)

Variables for the 2nd test:

All variables remained the same except for the following:

Number of Rows	1002
----------------	------

Results: Average Absolute Error = 1.3347 (Low Error)

Answer: Between the two test the error did improve very much so, it is most likely that Aim did not need very many more rows to learn this particular function. We then brought up the question, "How many rows can AIM import ?" We ran another test to find out.

Variables for the 3rd test:

All variables remained the same except for the following:

Number of Rows	30,002
----------------	--------

Results: Aim froze

Answer: AIM could import the data but could not create tables for the data. Abtech's software engineer told us that the largest array Aim could handle was 100 columns by 8000 rows.

Limits

Problem: What is the result of setting limits on the function AIM is trying to learn ?

Variables for the 1st test (no limits):

Data Generation	C program
Polynomial	$y = (c1) * (x1^2+c2) * (x1) * (x2+c3) * (x2^2+c4)$
Coefficients	c1 = random number between 2500 and -2500 c2 = random number between 2500 and -2500 c3 = random number between 2500 and -2500 c4 = random number between 2500 and -2500
Number of Row	500
Input Range	2500 and -2500
CPM	1
Root location	none specified
Analysis	Based results on Average Absolute Error
Results:	Average Absolute Error = 1.3663e+008 (High Error)

Variables for the 2nd test (with headers):

Polynomial	$y = (c0+c1) * (x1+c2) * (x1^2+c3) * (x1^3)$
Coefficients	c0 = 1 c1 = 9 c2 = 7 c3 = 8
Number of Rows	1002
Input Range	20 and -20
CPM	1
Root location	21 and -21

Results: Average Absolute Error = 2.2486e-013 (Low Error)

Answer: Positive improvement. The reason we needed these "headers" was because we needed to specify exactly where the ends of the function were, so that AIM did not take an area where there was no data and try to fit that into the curve (see figure 1-1, 1-2) ultimately smoothing the function. When generating random numbers you don't often create the maximum or minimum number for the range you set for generation.

figure1-1

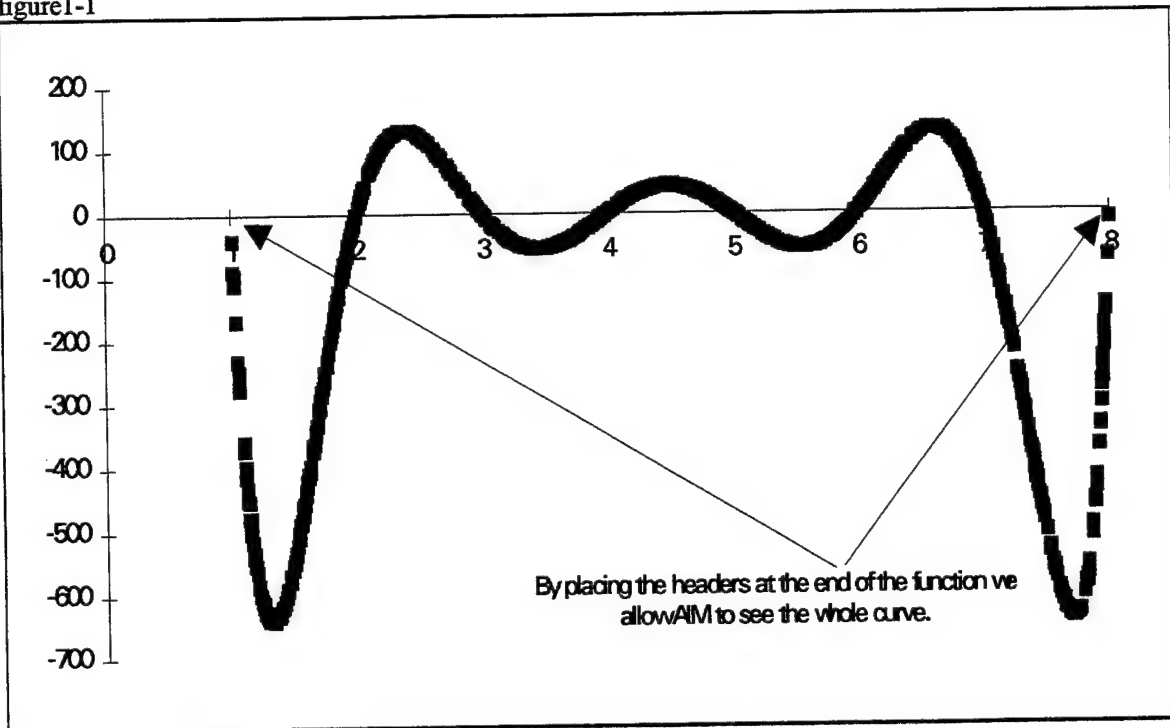
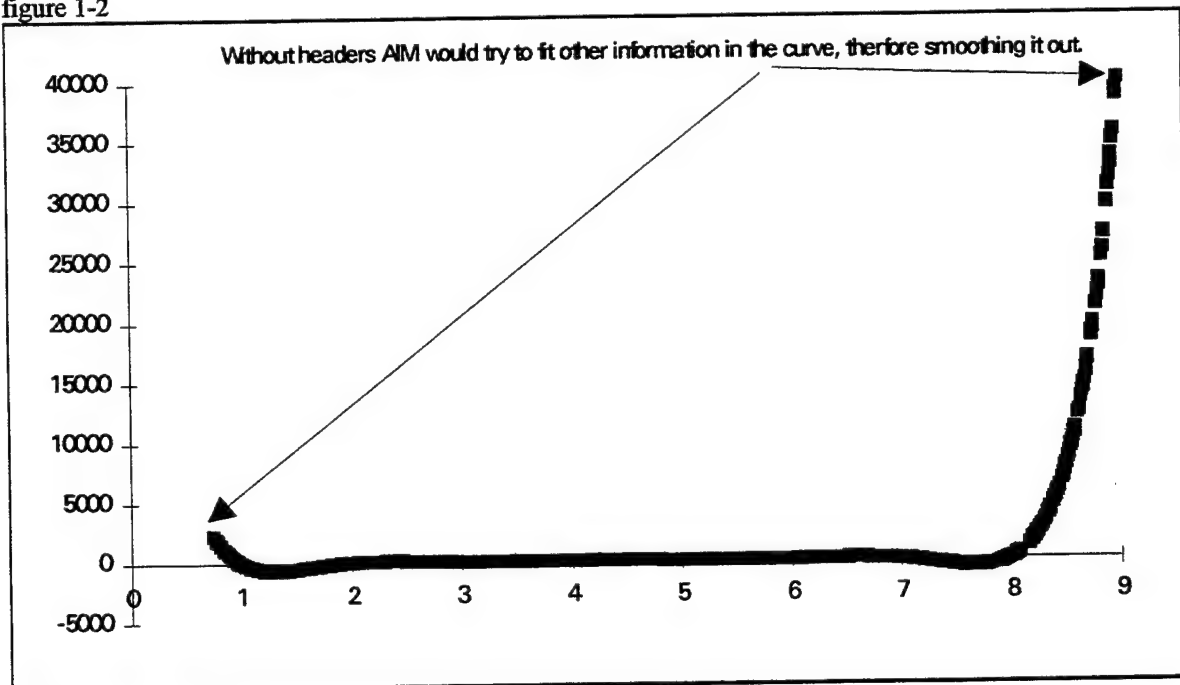


figure 1-2



Range

Problem: Does changing the range in which the input is sampled from have an effect on AIM's results ?

Variables for the 1st test:

Data Generation	Used a C program
Polynomial	$y = c_0 + c_1 * x_1 + c_2 * x_1^2 + c_3 * x_1^3$
Coefficients	$c_0 = 1$
	$c_1 = 9$
	$c_2 = 7$
	$c_3 = 8$
Number of Rows	1002
Input Range	1000 to -1000
CPM	1
Root location	1100 to -1100
Analysis	Based results on Average Absolute error

Results: Average Absolute Error = 8.1913e+005 (High Error)

Variables for the 2nd test:

All the variables remained the same except for the following:

Input Range	20 to -20
Root location	21 to -21

Results: Average Absolute Error = 344.79 (Average Error)

Variables for the 3rd test:

All the variables remained the same except for the following:

Input Range	1 to -1
Root location	2 to -2

Results: Average Absolute Error = 2.2486e-013 (Low Error)

Answer: The first two test produced large errors, but the third test was the error we were looking for. By looking at the reasonable size differential in the error and the range of the input, we concluded that in order for AIM to learn a function well, the input data must be clustered around the function of the equation.

CPM

Problem: What effect does a lower CPM (Complex Penalty Multiplier) have on AIM's results ?

Variables for the 1st test:

Data Generation	C program
Polynomial	$y = (c0+c1) * (x1+c2) * (x1^2+c3) * (x1^3)$
Coefficients	$c0 = 1$
	$c1 = 9$
	$c2 = 7$
	$c3 = 8$
Number of Rows	1002
Input Range	1000 to 1000
CPM	1
Root location	1100 to -1100
Analysis	Based results on Average Absolute Error

Results: Average Absolute Error = 8.1913e+005 (High Error)

Variables for the 2nd test (CPM changes):

All the variables remained the same except for the following:

CPM 0.01

Results: Average Absolute Error = 8.1913e+005 (High Error)

Answer: As you can see there was no obvious change in the results .Since the one variable equation was of low order, it created a simple network. After looking at larger order equation tests it was then determined that we would need to lower the CPM for a more complex equation in order to allow for a more complex network, resulting in a lower error.

Variables for the 3rd test:

Data Generation	Excel spread sheet
Polynomial	$y = (x+c1) * (x+c2) * (x+c3) * (x+c4) * (x+c5) * (x+c6) * (x+c7) * (x+c8) * (x+c9) * (x+c10) * (x+c11) * (x+c12) * (x+c13) * (x+c14) * (x+c15)$
Coefficients	$c1 = -1$ $c6 = -6$ $c11 = -11$
	$c2 = -2$ $c7 = -7$ $c12 = -12$
	$c3 = -3$ $c8 = -8$ $c13 = -13$
	$c4 = -4$ $c9 = -9$ $c14 = -14$
	$c5 = -5$ $c10 = -10$ $c15 = -15$
Number of Rows	1002
Input range	1 to 15.2
CPM	1
Root location	1 to 15.2
Analysis	Based results on Average Absolute Error

Results: Average Absolute Error = 1.969e+008 (High Error)

Variables for the 4th test (CPM):

All the variables remained the same except the following:

CPM 0.01

Results: Average Absolute Error = 1.969e+008 (High Error)

Answer: As you can see the Cpm had no affect on the results.

Mapped Variable Testing

After looking at all the changes that we needed to make in our data from previous testing, we wanted to integrate those changes into our next sets of tests. We decided to map out a route to take with the polynomials. We started with a simple, one variable, first order polynomial ($y = x-1$). From there we would go to a higher polynomial and see if there was a point at which the software would crack under the one variable case. After finding that point, we would move on to a second variable case and try to find the order of that set in which the software would crack. And then from there, on to the third variable case and so on until there was a point in which the software could not correctly calculate a first order of the highest variable case possible.

Figure 2-1 shows the exact route of the orders and variables we took
figure 2-1

Numbers of Variables								
Orders	1 var.	2 var.	3 var.	4var	5var	6var	7var	8var
1	1	1-1	1-1-1	3-3-3-3	3^5	3^6	3^7	3^8
2		1-2	1-1-2					
3		1-3	1-1-3					
4		2-2	1-1-5					
5		3-3	1-1-10					
6		4-2	2-2-2					
7		4-4	3-3-3					
8		5-5	5-5-5					
10								
15								
20								

1 Variables Test

Problem: At what point will AIM no longer be able to correctly calculate the solution for the one variable input?

Variables for the 1st test:

Data Generation	Used an Excel spread sheet
Polynomial	$y = (x-1)$
Number of Rows	1002
Range for input	.75 to 1.25
CPM	1
Root location	.75 to 1.25
Analysis	Based results on Normalized mean squared error and

comparison with other polynomials and error statistics.

Results: Normalized mean squared Error = 6.16e-009 (Low Error)

Variables for the 2nd test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2)$
Input Range	.75 to 2.25
Root location	.75 to 2.25

Results: Normalized mean squared Error = 4.37e-08 (low Error)

Variables for the 3rd test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3)$
Input Range	.75 to 3.25
Root location	.75 to 3.25

Results: Normalized mean squared Error = 3.27e-06 (low Error)

Variables for the 4th test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4)$
Range for input	.75 to 4.25
Root location	.75 to 4.25

Results: Normalized mean squared Error = 0.00022818 (low Error)

Variables for the 5th test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4) * (x-5)$
Range for input	.75 to 5.25
Root location	0 to 6

Results: Normalized mean squared Error = 1.46e-06 (low Error)

Variables for the 6th test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4) * (x-5) * (x-6)$
------------	---

Input Range	.75 to 6.25
Root location	0 to 7

Results: Normalized mean squared Error = 6.41e-13 (low Error)

Variables for the 7th test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4) * (x-5) * (x-6) * (x-7)$
Input Range	.9 to 7.1
Root location	.9 to 7.1

Results: Normalized mean squared Error = 0.030858 (low Error)

Variables for the 8th test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4) * (x-5) * (x-6) * (x-7) * (x-8)$
Input Range	1 to 7.25
Root location	1 to 7.25

Results: Normalized mean squared Error = 0.00010886 (low Error)

Variables for the 10th test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4) * (x-5) * (x-6) * (x-7) * (x-8) * (x-9) * (x-10)$
Input Range	1 to 9.25
Root location	1 to 9.25

Results: Normalized mean squared Error = 6.66e-03 (low Error)

Variables for the 15th test:

All the variables remained the same except for the following:

polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4) * (x-5) * (x-6) * (x-7) * (x-8) * (x-9) * (x-10) * (x-11) * (x-12) * (x-13) * (x-14) * (x-15)$
Input Range	1 to 15.1
Root location	1 to 15.1

Results: Normalized mean squared Error = 6.81e-03 (low Error)

Variables for the 20th test:

All the variables remained the same except for the following:

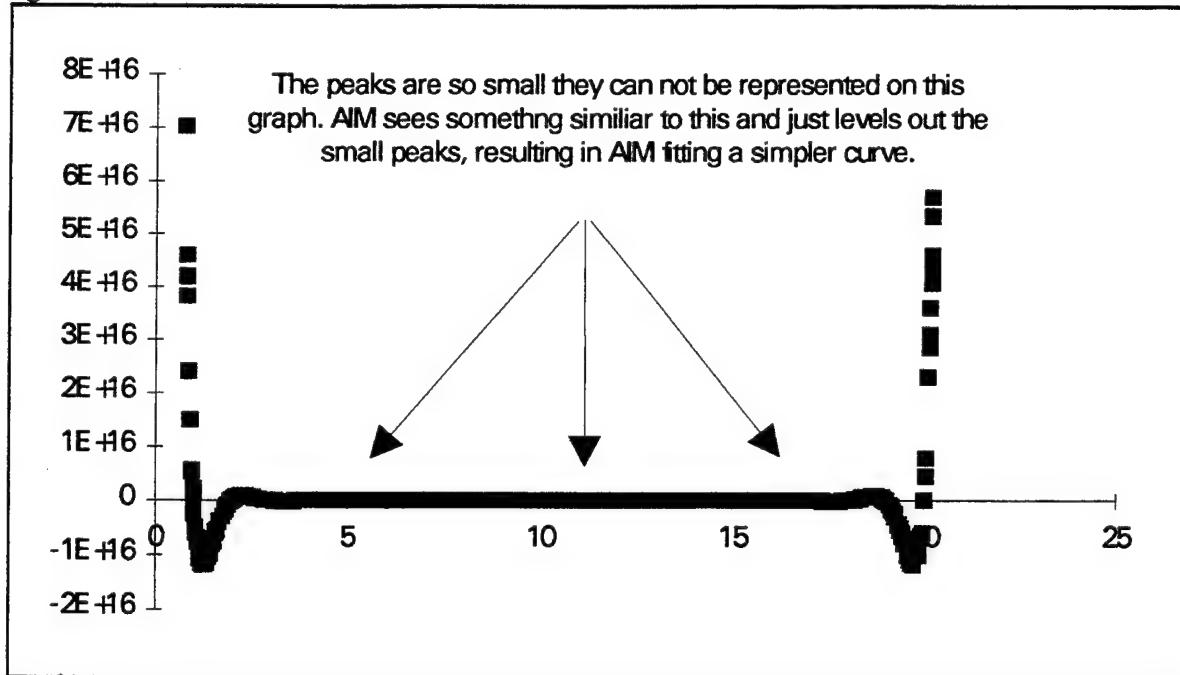
polynomial	$y = (x-1) * (x-2) * (x-3) * (x-4) * (x-5) * (x-6) * (x-7) * (x-8) * (x-9) * (x-10) * (x-11) * (x-12) * (x-13) * (x-14) * (x-15) * (x-16) * (x-17) * (x-18) * (x-19) * (x-20)$
Input Range	.75 to 20.25
Root location	.75 to 20.25

Results: Normalized mean squared Error = 6.67e-03 (low Error)

Answer: There were no large errors, but notice the fluctuation in the errors. Look at the 15th order equation, and the 20th order equation. The 20th order equation has a better error than the 15, we have a theory as to why this is occurring. Looking at the 20th order equation, the natural curve should have 20 "peaks," 2 large peaks at the ends and 18 smaller peaks in between the larger ones. What we believe AIM is doing is taking the curve and only fitting the larger peaks on the ends and straightening out the peaks in the middle. This is most likely due to the fact that the peaks in the middle of the two larger ones are relatively smaller than the two on the end (see figure 3-1). So theoretically AIM thinks it is fitting a smaller curve, and is therefore producing better error, than the equations in which the "middle" peaks are larger and in less numbers. Due to the fact that there were no large errors and our lack of time we decided to move on to the second variable sets.

Finishing the one variable case we were very excited to see the results for the second variable test set. After programming the excel sheet for two variable tests we labeled the individual data files and began importing. At the point of importation we received an error message from Microsoft "[Microsoft] [ODBC Single Tier Driver] Syntax Error" then after we received dismissed the window we received another error message window, this time from AIM, "Offending Statement : 'Create Table Data EP1-1(Id1 DOUBLEFLOAT, "Y" DOUBLEFLOAT, "X1" DOUBLEFLOAT, "X2" DOUBLEFLOAT). What this window was basically telling us was that AIM could not create the data tables for EP1-1, which was the file name for our raw data. This put a halt on things, because we wanted to test as much as we could in the time we had left, we tried changing the format in which the file was saved, (Tab separated to CSV), no effect. Finally we sent a copy of the file to Abtech to see if they could tell us what had happened. Meanwhile we kept on trying to find ways to make AIM import the data, our plan was to start with a simple file that AIM could import, and slowly mold it into a file we could use for testing. These test were imported one after another, until finally we thought that the reason AIM was not importing our actual test file was due to the file name. The names we were using for the simple files were TMP1, TMP2..., the names we were using for our actual test were EP1-1, EP1-2...etc. For some reason, AIM could not import filenames with hyphens. We renamed the files accordingly and the data was imported properly.

figure 3-1



2Variable test

Problem: At what point will no longer be able to correctly calculate the solution for the two variable input?

Variables for the 1st test:

Data Generation	Used an Excel spread sheet
Polynomial	$y = (x1-1) * (x2-2)$
Number of Rows	1002
input Range x1	.75 to 1.25
Input Range x2	.75 to 2.25
CPM	1
Root location x1	.75 to 1.25
Root location x2	.75 to 2.25
Analysis	Based results on Normalized mean squared error and comparison with other polynomials and error statistics.

Results: Normalized mean squared Error = 3.81e-04 (Low Error)

Variables for the 2nd test:

All the variables remained the same as the 1st test except for the following:

polynomial	$y = (x1-1) * (x2-2) * (x2-3)$
Input Range x2	.75 to 3.25
Root location x2	.75 to 3.25

Results: Normalized mean squared Error = 2.56e-04 (low Error)

Variables for the 3rd test:

All the variables remained the same as the 2nd test except for the following:

polynomial	$y = (x1-1) * (x2-2) * (x2-3) * (x2-4)$
Input Range x2	.75 to 4.25
Root location x2	.75 to 4.25

Results: Normalized mean squared Error = 8.21e-03 (low Error)

Variables for the 4th test:

All the variables remained the same as the 3rd test except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x2-3) * (x2-4)$
Input Range x2	.75 to 2.25
Root location x2	.75 to 4.25

Results: Normalized mean squared Error = 0.0015503 (low Error)

Variables for the 5th test:

All the variables remained the same as the 4th variable test except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x1-4) * (x2-5) * (x2-6)$
Input Range x1	.75 to 3.25
Input Range x2	.75 to 6.25
Root location x1	.75 to 3.25
Root location x2	.75 to 6.25

Results: Normalized mean squared Error = 7.69e-03 (low Error)

Variables for the 6th test:

All the variables remained the same the 5th test except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x2-4) * (x2-5) * (x2-6)$
------------	---

Results: Normalized mean squared Error = 0.0047937 (low Error)

Variables for the 7th test:

All the variables remained the same as the 6th test except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x1-4) * (x2-5) * (x2-6) * (x2-7) * (x2-8)$
Input Range x1	.75 to 4.25

Input Range x2	4.75 to 8.25
Root location x1	.75 to 4.25
Root location x2	4.75 to 8.25

Results: Normalized mean squared Error = 0.0088914 (low Error)

Variables for the 8th test:

All the variables remained the same as the 7th test except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x1-4) * (x1-5) * (x2-6) * (x-7) * (x-8) * (x-9) * (x-10)$
Input Range x1	.75 to 5.25
Input Range x2	5.75 to 10.25
Root location x1	.75 to 3.25
Root location x2	5.75 to 10.25

Results: Normalized mean squared Error = 0.027927 (low Error)

Answer: AIM is getting the lower order equations quite well, but in the higher order equations the smoothing affect may be taking place.

3 Variable test

Problem: Will AIM continue to become inaccurate as the 3rd variable testing progresses?

Variables for the 1st test:

Data Generation	Used an Excel spread sheet
Polynomial	$y = (x1-1) * (x2-2) * (x3-3)$
Number of Rows	1002
input Range x1	.75 to 1.25
Input Range x2	.75 to 2.25
Input Range x3	2.75 to 3.25
CPM	1
Root location x1	.75 to 1.25
Root location x2	.75 to 2.25
Root location x3	2.75 to 3.25
Analysis	Based results on Normalized mean squared error and comparison with other polynomials and error statistics.

Results: Normalized mean squared Error = 8.78e-04 (Low Error)

Variables for the 2nd test:

All the variables remained the same as the 1st test except for the following:

polynomial	$y = (x1-1) * (x2-2) * (x3-3) * (x3-4)$
------------	---

Results: Normalized mean squared Error = 1.2e-01 (low Error)

Variables for the 3rd test:

All the variables remained the same as the 2nd test except for the following:

polynomial	$y = (x1-1) * (x2-2) * (x3-3) * (x3-4) * (x3-5)$
Input Range x3	2.75 to 5.25
Root location x3	2.75 to 3.25

Results: Normalized mean squared Error = 4.89e-02 (low Error)

Variables for the 4th test:

All the variables remained the same as the 3rd test except for the following:

polynomial	$y = (x1-1) * (x2-2) * (x3-3) * (x3-4) * (x3-5) * (x3-6) * (x3-7)$
Input Range x3	2.75 to 7.25
Root location x3	2.75 to 3.25

Results: Normalized mean squared Error = 0.57555 (low Error)

Variables for the 5th test:

All the variables remained the same as the 4th test except for the following:

polynomial	$y = (x1-1) * (x2-2) * (x3-3) * (x3-4) * (x3-5) * (x3-6) * (x3-7) * (x3-8) * (x3-9) * (x3-10) * (x3-11) * (x3-12)$
Input Range x3	2.75 to 12.25
Root location x3	2.75 to 3.25

Results: Normalized mean squared Error = 8.21e-02 (low Error)

Variables for the 6th test:

All the variables remained the same as the 6th test except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x2-3) * (x2-4) * (x3-5) * (x3-6)$
Input Range x1	.75 to 2.25
Input Range x2	2.75 to 4.25
Input Range x3	4.75 to 6.25
Root location x1	.75 to 2.25
Root location x2	2.75 to 4.25
Root location x3	4.75 to 6.25

Results: Normalized mean squared Error = 0.21897 (low Error)

Variables for the 7th test:

All the variables remained the same except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x2-4) * (x2-5) * (x2-6) * (x3-7) * (x3-8) * (x3-9)$
Input Range x1	.75 to 3.25
Input Range x2	3.75 to 6.25
Input Range x3	6.75 to 9.25
Root location x1	.75 to 3.25
Root location x2	3.75 to 6.25
Root location x3	6.75 to 9.25

Results: Normalized mean squared Error = 1.0037 (low Error)

Variables for the 8th test:

All the variables remained the same except for the following:

polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x1-4) * (x1-5) * (x2-6) * (x2-7) * (x2-8) * (x2-9) * (x2-10) * (x3-11) * (x3-12) * (x3-13) * (x3-14) * (x3-15)$
Input Range x1	.75 to 5.25
Input Range x2	5.75 to 10.25
Input Range x3	10.75 to 15.25
Root location x1	.75 to 5.25
Root location x2	5.75 to 10.25
Root location x3	10.75 to 15.25

Results: Normalized mean squared Error = 1.3523 (Low Error)

Answer: There is fluctuation in the error, but none of the errors are extremely large. The fluctuation is probably due to the straightening out of smaller peaks (see figure 1-2) between larger ones.

Due to our lack of time, we only able to run one test for each of the next four variables and grouped them together in one test set.

4 Variable - 8 Variable tests

Problem: Will AIM continue to produce acceptable errors as the numbers of variable increase ?

Variables for the 4th variable test:

Data Generation	Used an Excel spread sheet
Polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x2-4) * (x2-5) * (x2-6) * (x3-7) * (x3-8) * (x3-9) * (x4-10) * (x4-11) * (x4-12)$
Number of Rows	1000
input Range	.75 to 3.25
Input Range x2	3.75 to 6.25
Input Range	6.75 to 9.25
Input Range x4	9.75 to 12.25
CPM	1

Root location x1	.75 to 3.25
Root location x2	3.75 to 6.25
Root location x3	6.75 to 9.25
Root location x4	9.75 to 12.25
Analysis	Based results on Normalized mean squared error and comparison with other polynomials and error statistics.

Results: Normalized mean squared Error = 1.18e+00 (Low Error)

Variables for the 5th variable test:

All variables remained the same as the 4th variable test except for the following:

Polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x2-4) * (x2-5) * (x2-6) * (x3-7) * (x3-8) * (x3-9) * (x4-10) * (x4-11) * (x4-12) * (x5-13) * (x5-14) * (x5-15)$
Input Range x5	12.75 to 15.25
Root location x5	12.75 to 15.25

Results: Normalized mean squared Error = 1.24E+00 (Low Error)

Variables for the 6th variable test:

All variables remained the same as the 5th variable test except for the following:

Polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x2-4) * (x2-5) * (x2-6) * (x3-7) * (x3-8) * (x3-9) * (x4-10) * (x4-11) * (x4-12) * (x5-13) * (x5-14) * (x5-15) * (x6-16) * (x6-17) * (x6-18)$
Input Range x6	15.75 to 18.25
Root location x6	15.75 to 18.25

Results: Normalized mean squared Error = 9.98e-01 (Low Error)

Variables for the 7th variable test:

All variables remained the same as the 6th variable test except for the following:

Polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x2-4) * (x2-5) * (x2-6) * (x3-7) * (x3-8) * (x3-9) * (x4-10) * (x4-11) * (x4-12) * (x5-13) * (x5-14) * (x5-15) * (x6-16) * (x6-17) * (x6-18) * (x7-19) * (x7-20) * (x7-21)$
Input Range x7	18.75 to 21.25
Root location x7	18.75 to 21.25

Results: Normalized mean squared Error = 1.0043 (Low Error)

Variables for the 8th variable test:

All variables remained the same as the 7th variable test except for the following:

Polynomial	$y = (x1-1) * (x1-2) * (x1-3) * (x2-4) * (x2-5) * (x2-6) * (x3-7) *$
------------	--

	$(x3-8) * (x3-9) * (x4-10) * (x4-11) * (x4-12) * (x5-13) * (x5-14) * (x5-15) * (x6-16) * (x6-17) * (x6-18) * (x7-19) * (x7-20) * (x7-21) * (x8-22) * (x8-23) * (x8-24)$
Input Range x7	21.75 to 24.25
Root location x7	21.75 to 24.25

Results: Normalized mean squared Error = 1.26e+00 (Low Error)

Answer: The error for these tests, also did not follow a consistent pattern of decreasing chronologically. However, looking at the overall error, it seems to be following a slow path of becoming larger.

Result Summary

We did not get to extend our testing as far as we would like but we did find some interesting results. Looking at the tests in which we examined different variables for the input we know four different methods for improving the results.

1. **The number of rows.**

These rows are examples that trains itself on and learns from. Adding more rows will give Aim a larger sample of the function you want it to learn.

2. **The limits (headers).**

By adding these points you define the boundaries for the function outside the boundaries of the random sampling. So that way Aim is not looking at just a small part of the function or more than the function, but just outside the function. (see figures 1-1)

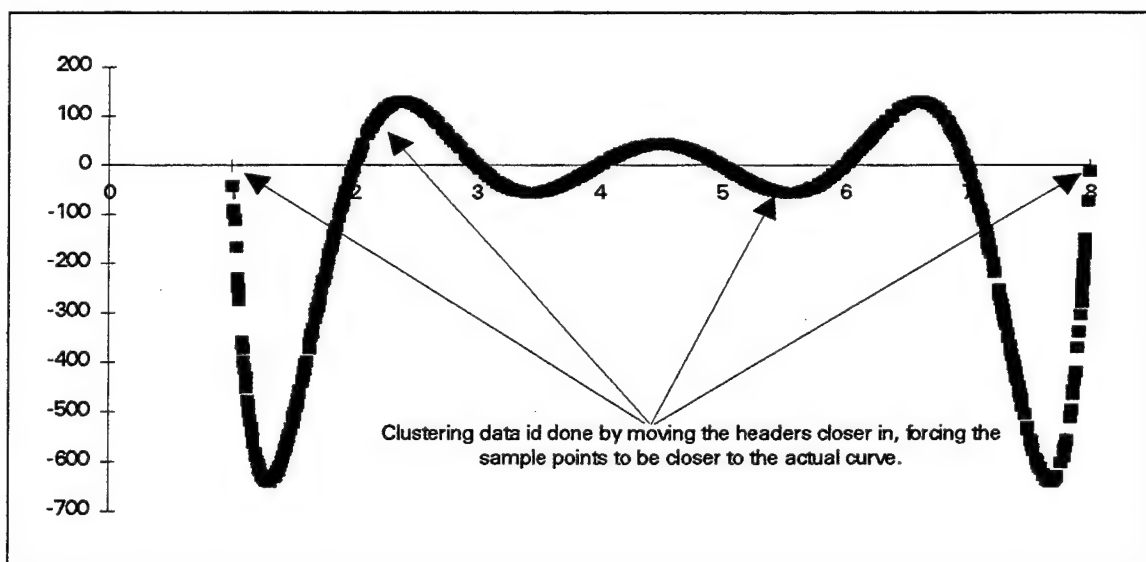
3. **The range.**

One of the major testing break through this summer was when we discovered that in order for the learned roots or the projection of the roots to look like the roots (input) you need to cluster your data points very closely to the function(see figure 3-2)

4. **CPM**

When dealing with a complex equation, in theory, lowering the CPM should allow for a more complex network. In our testing of the CPM the changing of the CPM had no apparent affect on the results. This may be due to the simplicity of the networks examined.

figure3-2



Looking at the tables for our mapped out testing we see how AIM begins to drift slowly towards a large error then pulls back, this maybe due to the smoothing affect. But further testing needs to be completed to confirm this theory.

Var No.	Ag. Absl Error	Nmrld Mean Sq Error	DBase Max	DBase Min	Var No.	Orders	Ag. Absl Error	Nmrld Mean Sq Error	DBase Max	DBase Min
1	1.70E-06	616E-09	0.39744	-0.19938	3	1_1_1	8.78E-04	2.08E-03	0.18993	-0.16675
	4.27E-06	4.37E-09	0.31241	-0.24998	1	1_1_2	9.59E-03	1.12E-01	0.012327	-0.01235
	4.66E-04	3.27E-06	0.67549	-0.64433	1	1_1_3	2.32E-03	4.89E-02	0.076583	-0.11408
	0.0085133	0.0002818	2.1733	-0.99993	1	1_1_5	0.024009	0.57555	0.16681	-0.25381
	0.027342	1.46E-06	113.65	-117.78	1	1_1_10	344.62	8.21E-02	33570	-14546
	4.16E-05	6.41E-13	674.01	-16.827	2	2_2_2	1.94E-03	2.19E-01	0.018185	-0.02008
	6.1602	0.03888	95.841	-95.778	3	3_3_3	0.019707	1.0087	0.16411	-0.11003
	1.384	0.00010886	129.74	-640.28	5	5_5_5	8.36E-40	1.35E-40	125.82	-118.27
	51246	6.66E-03	6554.7	-42545	4	3_3_3_3	5.03E-04	1.18E-40	0.019255	-0.02898
	1.67E-48	6.81E-03	2.52E-40	-9.10E-49	5	3_5_5	1.55E-03	1.24E-40	0.008548	-0.01905
	4.70E-44	6.67E-03	6.95E-46	-1.20E-46	6	3_6_6	3.47E-04	9.98E-01	0.005155	-0.00412
2	2.08E-03	3.81E-04	0.27366	-0.28224	7	3_7_7	0.0004778	1.0043	0.084663	-0.00204
	1.71E-03	2.55E-04	0.53874	-0.62727	8	3_8_8	2.50E-05	1.26E-40	0.000287	-0.0025
	1.17E-02	8.21E-03	1.3488	-2.2216						
	0.012719	0.0015803	2.0883	-1.7754						
	0.23822	7.68E-03	17.195	-22.585						
	0.25715	0.0047987	23.31	-26.89						
	14.935	0.0089914	23104	-937.71						
	23063	0.24154	54882	-54888						
	343E-44	0.094997	1.85E-46	-7.30E-45						

Conclusion

Examining the results we can say that AIM handles most sized equations fairly well. However, when dealing with larger or more complex equations AIM's error seem to be actually becoming better. We have looked at the results closely and have decided that AIM is most likely fitting a simpler curve by straightening out the smaller peaks in between the larger peaks. (see figures 3-1, 3-2) This why that in order to see how accurately AIM can calculate larger equations more testing needs to be completed and analyzed.

Future Studies

The testing on AIM is a very large unexplored area. There are so many different variables that random research would not be very effective. Some suggested research would be areas that created the guidelines for this research, such as:

- Going back and redoing the CPM for the 1 variable tests of 10, 15, 20. We found out that it may be possible to raise the CPM and get better result, but this is only for complex equations.
- Determine the function for the mapped out equation.
- Compare the curves, to see the difference in each.

Some other areas of testing for independent studying could be:

CPM - Examine the affect of the CPM for polynomials of other complexities, for a broader range of CPM values, and for the number of training samples.

Range - Look at placing the roots at different locations (especially on the 2 variable or highly complex equation)

Rows - Examine the affects of giving AIM the lowest possible rows or maximum possible rows over different size equations.

Coefficients - Experiment the capability of AIM to produce excellent errors with different coefficients.

Number of variables - Examine the "route" of equations in which AIM begins to break down (see figure 2-1)

Variables - Independent study on any of the variables AIM provides. Examples: Operations, variables you can set that can extract more information from data. Synthesis of settings, these variables can allow for a more complex network, resulting in lower error. Generation, the variables in generation are ones that determine how AIM splits and creates its own data.

All of these variables change AIM's final results. AIM is equipped with many different variables in different aspects of the data; generating, synthesizing, and evaluating, it would seem that Aim has a specific way to handle different problems. Because we did not know the specifications for each test, not every error looked liked it could have. Our testing has only sampled the vast area of AIM's capabilities, and this is why similar testing in this area is encouraged.

References

1. **Users's Manual AIM for Windows Version 2.0**
Abtech Corporation, Charlottesville, VA

**ELEVATED TEMPERATURE CHARACTERISTICS
OF A SILICON DIODE**

Angela C. Rabe

**Carroll High School
4524 Linden Ave.
Dayton, OH 45432**

**Final Report for:
High School Apprentice Program
Wright Laboratory**

**Sponsored By:
Air Force Office of Scientific Research
Bolling Air Force Base, DC**

and

Wright Laboratory

August 1995

ELEVATED TEMPERATURE CHARACTERISTICS OF A SILICON DIODE

Angela C. Rabe
Carroll High School

ABSTRACT

The effects of elevated temperature on a silicon diode were tested. The diode was heated in a vacuum chamber, and the forward and reverse characteristics were tested at different temperatures. The temperatures tested were the case (of the diode) temperatures starting at 24C and increasing in increments of 25C until 149C. After all of the current versus voltage data was graphed, the relationship between the current and voltage and the effect of temperature were determined, as well as an analysis of the relationship between the lead and case temperatures and the relationship between the $\ln(I)$ and $1/T$.

The case and lead temperatures were found to differ in magnitude due to the varying ways by which they received the additional heat. As far as the relationship between the $\ln(I)$ and $1/T$ is concerned, they were related through a fourth power curve.

From here it was determined that the temperature had a definite impact on the effectiveness of the diode. In the forward bias, for small positive voltages, the higher temperature settings produced more output current than the lower temperature settings at the same voltage, up to the point of conduction. For larger voltages, the trend reverses, due to increased internal resistance, and for a constant current, higher temperatures will require additional voltage to reach the same current. In the reverse bias condition, for higher temperatures the breakdown becomes more gradual. Before breakdown occurs, the higher temperature settings produce a greater absolute value of output current than the lower temperature settings for the same relatively low voltage. After the breakdown occurs, the lower temperature settings require a smaller absolute value of voltage in order to attain the same output current value as the lower temperatures.

ELEVATED TEMPERATURE CHARACTERISTICS OF A SILICON DIODE

Angela C. Rabe
Carroll High School

INTRODUCTION

There exists a vast variety of electronic systems in use at this time, all of which rely on the semiconductor diode at their foundation. These diodes are generally made from one of two semiconductor materials: Germanium (Ge) and Silicon (Si). They each consist of atomic structures that are tetravalent, with four electrons in their valence shell. These elements are used for a variety of reasons. They can be manufactured to a very high purity level and doped as needed. The existence of periodic patterns as well as the single crystal structure in these two elements is conducive to such purity levels. Their characteristics are also capable of being altered significantly by the application of heat or light.

As the temperature rises in a semiconductor material, an increasing number of electrons break their covalent bonds, with the help of absorbed thermal energy, and become free electrons. The negative temperature coefficient of resistance found in such materials as Si and Ge represents this reduction in resistance accompanying an increase in temperature. Resistance in most conductors, on the other hand, increases with temperature.

Although the number of carriers does not increase, their vibration within their energy levels makes it more difficult for electrons to pass through the p-n junction. The increase in temperature, for conductors, leads to an increased resistance level and a positive temperature coefficient.

An extrinsic material is one that has been doped, or greatly changed by the addition of impurity atoms to a relatively pure semiconductor material. There are two main divisions of extrinsic material. N-type material is created when impurity atoms containing five valence electrons are added to a relatively pure semiconductor material. These are referred to as donor atoms because they donate their fifth electron to the conduction layer, although the material still remains neutral as a whole. The addition of these electrons to the conduction layer increases the conductivity of the material. The other main division is p-type material, which is formed by adding impurity atoms with only three valence electrons to a relatively pure semiconductor material. This addition creates holes in the crystalline structure because there are not enough electrons to fill all of the covalent

bonds. These atoms are referred to as acceptors because they are willing to accept "free" electrons. The conventional flow is indicated by the direction of the hole flow. In the n-type material, the electrons greatly outnumber the holes. In this case, the electrons are the majority carriers and the holes are referred to as minority carriers. The situation is exactly the opposite for p-type material.

The ideal diode is a two-terminal device, consisting of a p-n junction. This is a joining of one p-type and one n-type material. When applying voltage, if the positive terminal is joined to the n-type material and the negative terminal is connected to the p-type material, then the depletion region will widen. The number of minority carriers entering the depletion region will not change, nor will the magnitude of the minority carrier flow vectors, from a case involving no applied voltage. The current in the case of this applied voltage is referred to as the reverse saturation current. In forward bias, the positive terminal is connected to the p-type material and the negative terminal is fused to the n-type material. This situation greatly reduces the depletion region and results in an exponentially increasing flow of majority carriers across the junction. The forward bias is represented in the first quadrant, when graphed; whereas, the reverse bias is depicted in the third quadrant.

At the junction there is a gradual change in electron and hole concentrations. These concentration gradients create diffusion fluxes, or currents, which leave unfulfilled immobile ionized atoms. This condition brings rise to drift currents that serve to counteract the diffusion

currents. There is also an energy barrier formed by the potential difference across the depletion region. Most of the n- and p-type material lying outside of the depletion region consists of almost all diffusion current.

In the reverse bias condition, a breakdown occurs at large negative voltages for one of two reasons. The Zener breakdown, named after the man who first discovered it, occurs when the valence band electrons tunnel into the empty conduction band states of the same energy under the influence of large electric fields. After a certain voltage, the filled states in the valence band of the p-region will have the same energy as the unoccupied states in the n-region on the opposite side of the junction. These areas are separated by a triangle-shaped potential barrier. The height of this barrier is equal in energy to the bandgap, and its thickness at small reverse voltages is so large that the probability of tunneling between states of equal energy is negligible. At small reverse voltages only the reverse saturation current flows. For larger reverse voltages, the slope of the triangular barrier is much steeper and the states of equal energy in the valence and conduction bands on opposite sides are separated by a much thinner barrier. This factor increases the tunneling probability significantly and the tunneling current increases the reverse current. At a certain reverse voltage, the tunneling current overcomes the saturation current and breakdown occurs.

The avalanche process is the second way that breakdown may occur. In this method, a free electron or hole acquires enough energy

from the electric field during the collision to generate an electron hole pair in collision with a lattice phonon. A phonon is the smallest measurement of atomic vibrations in a crystal lattice. The collisions that destroy the drift velocity of electrons occur between electrons and phonons, or between electrons and impurity atoms or lattice imperfections. In a pure crystalline material, most of the collisions occur between electrons and phonons. Since phonons have no charge, the conservation of momentum does not conserve current density. The number of available phonons increases as the temperature rises, and the electron-phonon collisions occur more frequently at higher temperatures. As the time interval between collisions decreases with increasing temperature, the electrical conductivity of normal metals decreases. In the avalanche process the free minority carriers develop a sufficient velocity to liberate additional carriers through ionization. For this to occur, the energy in the mean free

path must be equal to the bandgap energy. This avalanche process causes a breakdown when the additional carriers start a chain reaction, liberating even more carriers.

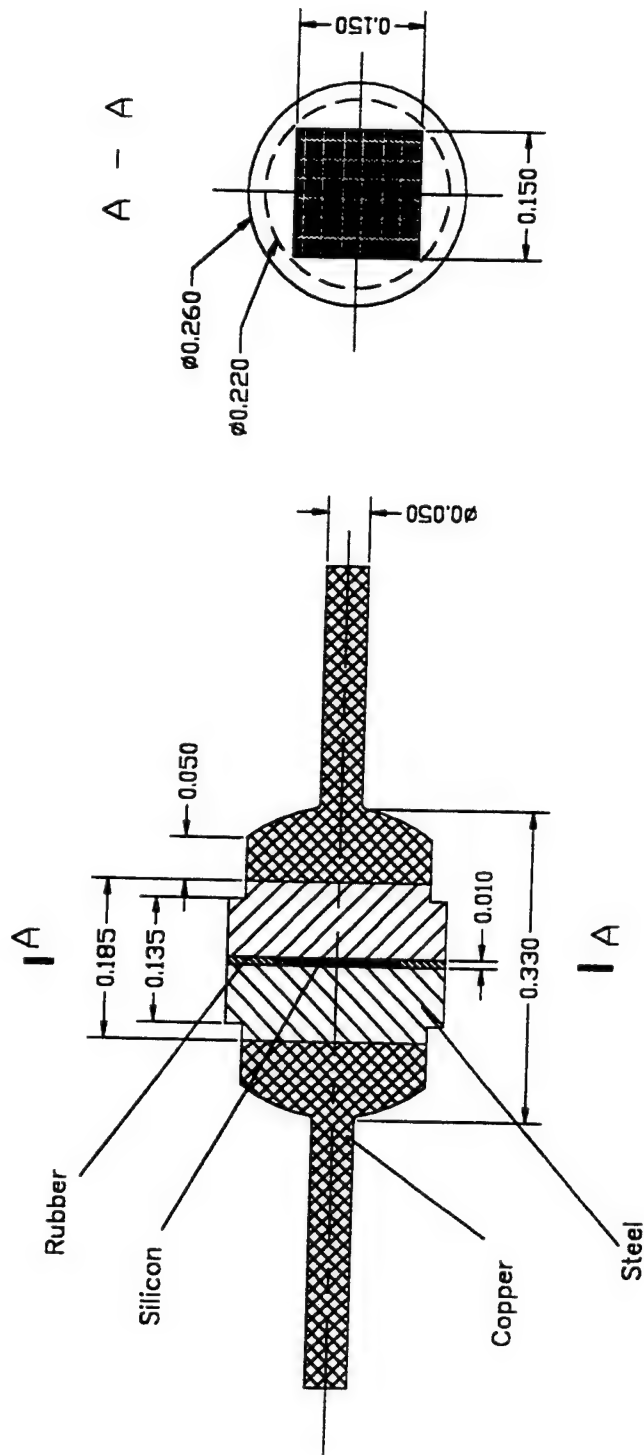
This experiment explores these basic characteristics of forward and reverse bias for a silicon diode, the specifications for which are provided in Table 1.

SETUP

The silicon diode with the structure shown in Fig. 1, was used in this experiment in order to examine case temperatures up to 150C, nearing the maximum operable value for silicon (200C), and exceeding such a value for other semiconductor materials. For the setup structure, the silicon diode was placed inside a high vacuum chamber with the case resting on a heating stage. The vacuum pump evacuates the air that may otherwise alter the data due to oxide formations, and the heater stage provides

Table 1: Specifications for an MR750 Silicon Diode

Peak Repetitive Reverse Voltage	50 Volts
Non-Repetitive Peak Reverse Voltage	60 Volts
RMS Reverse Voltage	35 Volts
Average Rectified Forward Current	22 Amps (TL=60C, 1/8" Lead Lengths)
Non-Repetitive Peak Surge Current	400 Amps (for 1 cycle)
Operating and Storage Junction Temperature Range	-65C to +175C
Maximum Instantaneous Forward Voltage	1.25 Volts (at 25C)
Maximum Forward Voltage Drop	0.90 Volts (at 25C)
Maximum Reverse Current	25 μ A (at 25C)



37-6

Fig. 1

an effective means of changing the temperature of the diode case. Three thermocouples were used to monitor various temperatures, with one on the case, one inside the heater stage, and one on a lead. The case and lead thermocouples were connected to separate temperature monitors, and the heater stage thermocouple was joined to a temperature controller. See Fig. 2 below for more detail.

The positive lead, the anode, was joined to the positive terminal on the curve tracer, and the negative lead, the cathode, was connected to the negative terminal, for the forward bias. The opposite connection was made for the reverse bias. From this connection, the data were immediately stored and graphed on the curve tracer and later transferred to the computer.

PROCEDURE

The temperature of the heater stage was varied so that data sets could be collected for case temperatures starting at room temperature and increasing in increments of about 25C and

ending at 149C as shown in Table 2. The lead temperatures were also measured at each heater case temperature to observe the relationship between case and lead temperatures.

For each case temperature, the curve tracer was set in the proper mode for forward bias. This included using the settings for high current and normal polarity. Data points were collected for current with increasing positive voltage. This means the voltage was applied through the positive lead. Then the setup was changed in order to collect the data for the reverse bias conditions at the same temperature. The voltage was then applied through the negative lead. In this case, the high voltage and reverse polarity were used, as opposed to the conditions for forward bias. For the reverse bias, data points were gathered for the changing current as the absolute value of the voltage increased. For these measurements, the voltage was applied through the cathode. At each temperature setting, both sets of data were collected, yielding twelve sets of data in all. The data sets were then transferred to the disk, graphed, and analyzed on the computer.

Table 2: Lead and heater temperatures for a given case temperature (C)

CASE TEMPERATURE	LEAD TEMPERATURE	HEATER TEMPERATURE
24	23	24
50	46	145
75	71	186
99	92	213
125	116	321
149	139	374

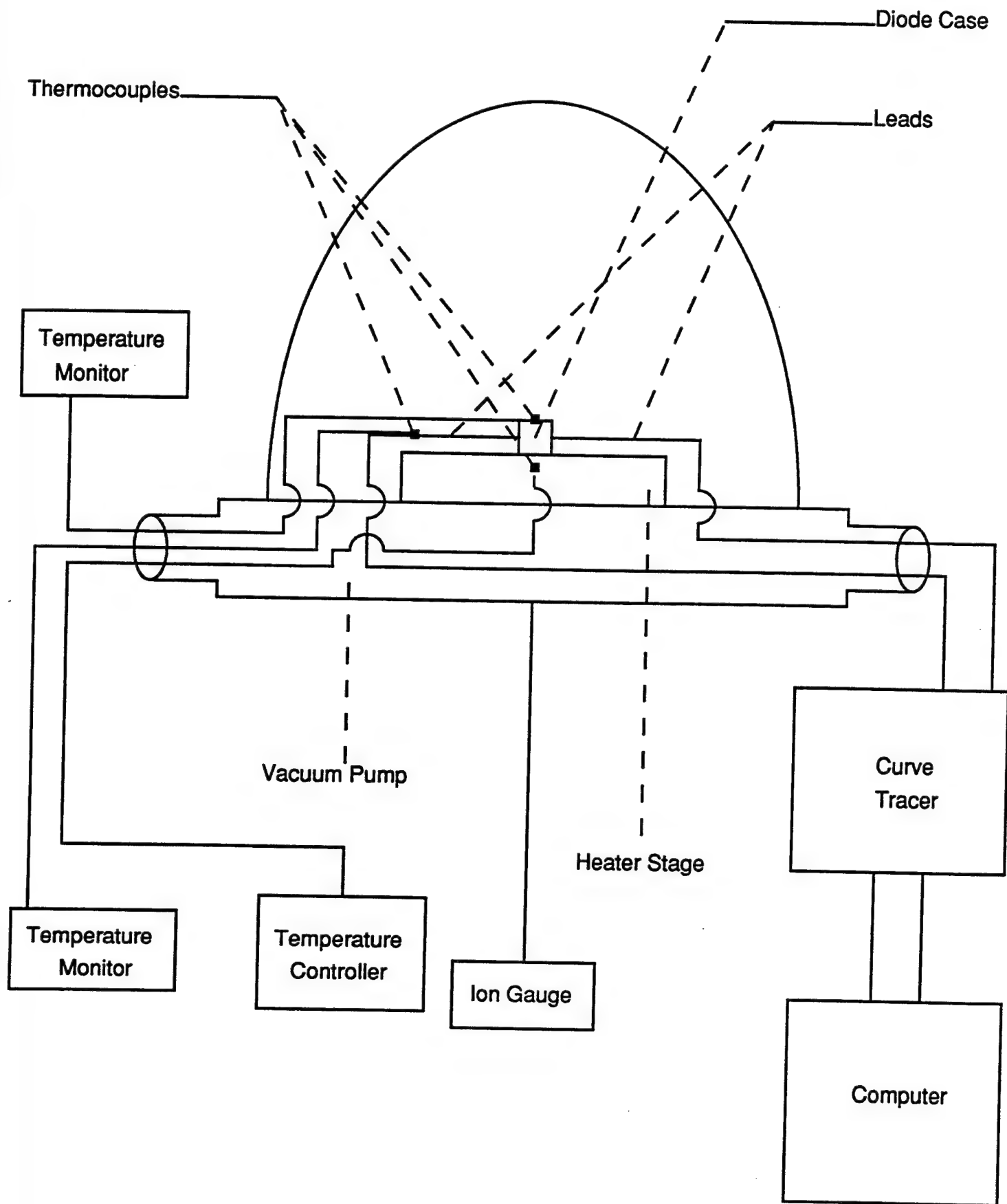


FIG. 2: A SCHEMATIC DIAGRAM OF THE SETUP

ANALYSIS

For the forward diode characteristics, shown in Fig. 3, the graphs show that with a constant current, just before the region of conduction, higher temperatures require less voltage to reach the same current. This trend continues until about 0.75 volts, the minimum voltage at which the diode starts conducting. Between 0.75 volts and almost 3.5 volts a transition period takes place as the diode conducts in the forward mode. Through extrapolation of the data, it can be shown that after 3.5 volts the trend has reversed, and for a constant current, higher temperatures will require additional voltage to reach the same levels of current. This can be explained by the vibration of the atoms, that generally results in an increased internal resistance of the diode. All atoms vibrate, but they can vibrate at various magnitudes. The greater the temperature of the atom, the larger the magnitude of the vibrations becomes, making it increasingly difficult for electrons to pass through the region of the junction. At this point the semiconductor material behaves as a conductor for these large applied voltages, causing an increase in temperature to lead to an increase in resistance. Due to the equation $V=IR$, maintaining the same current means even more voltage must be applied. This leads to a lower output current for higher temperature settings as opposed to that for lower temperature settings, for a constant voltage. This is the same as stating that for a constant current, higher temperatures will

require additional voltage to reach the same values of current.

It can also be noted that as the temperature rises, the voltage required to reach the region of conduction is less. This trend is to be expected, because in order to reach the region of conduction, the electrons must first acquire enough energy to be able to travel over the energy barrier. This additional energy may be gained through one of two means. The electrons may be given an increased applied voltage or an increased temperature. Therefore, the extra energy supplied by the higher temperatures allows the electrons to cross the energy barrier much sooner, resulting in larger output currents for a constant voltage. This behavior also indicates that due to an increase in the internal resistance at higher temperature, the voltage drop in the forward mode increases, thereby causing conduction at higher voltages.

The trend begins to change at approximately 0.75 volts. The reason such a change is possible, is that the energy barrier is approximately 0.75 volts. After all of the electrons have reached the required energy level by increased voltage, their temperature differences will indicate a different trend. The energy barrier is no longer a factor after all of the electrons are at 0.75 volts. These trends are illustrated in Figs. 3 and 4.

The reverse diode characteristics illustrate their own unique trends, as shown in Fig. 5. In general, as the temperature increases, the breakdown is more gradual and less rigid. The data for the 75C setup do not coincide with this trend. However, since the other five curves

DIODE FORWARD CHARACTERISTICS

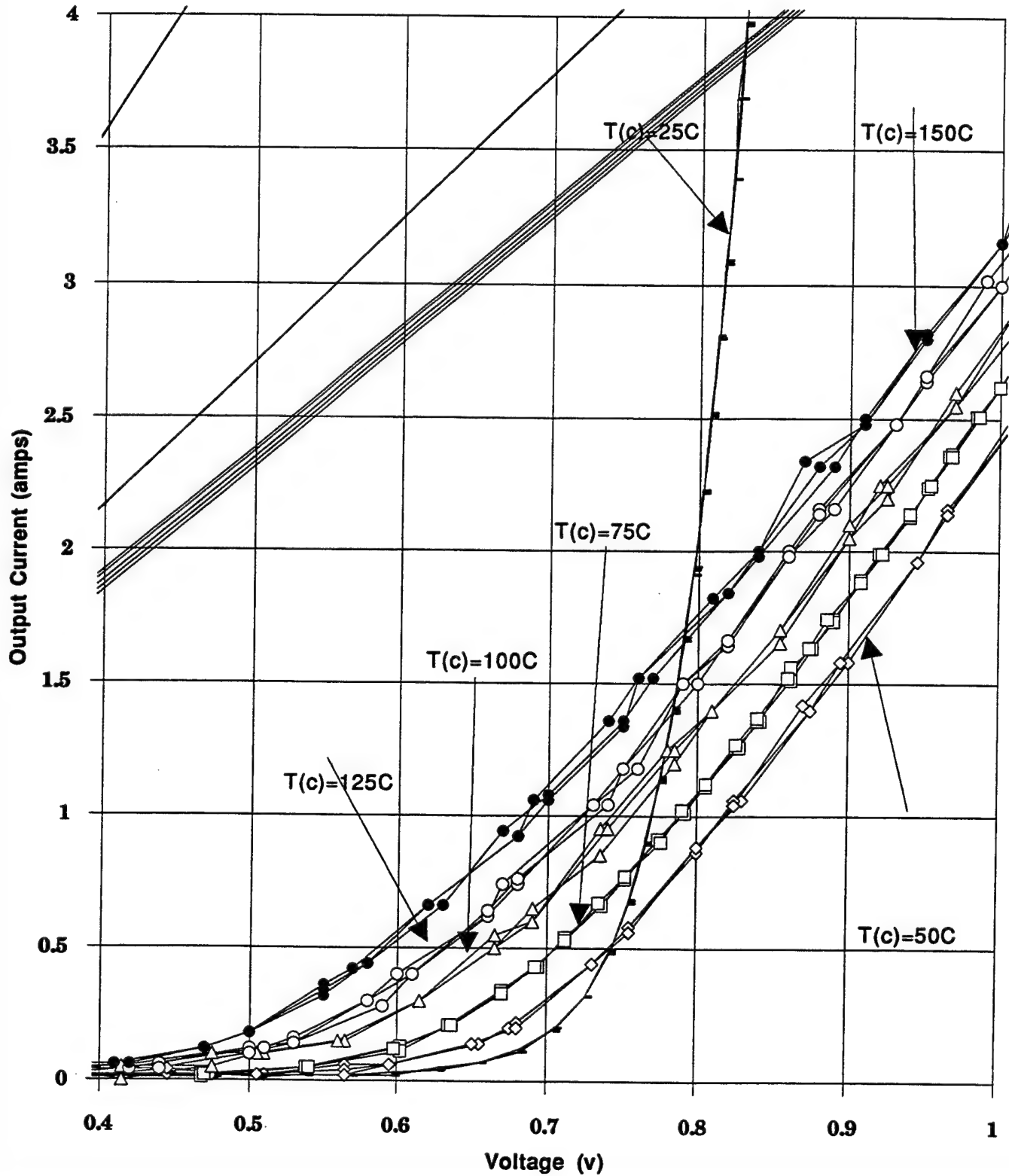


Fig. 3

DIODE FORWARD CHARACTERISTICS

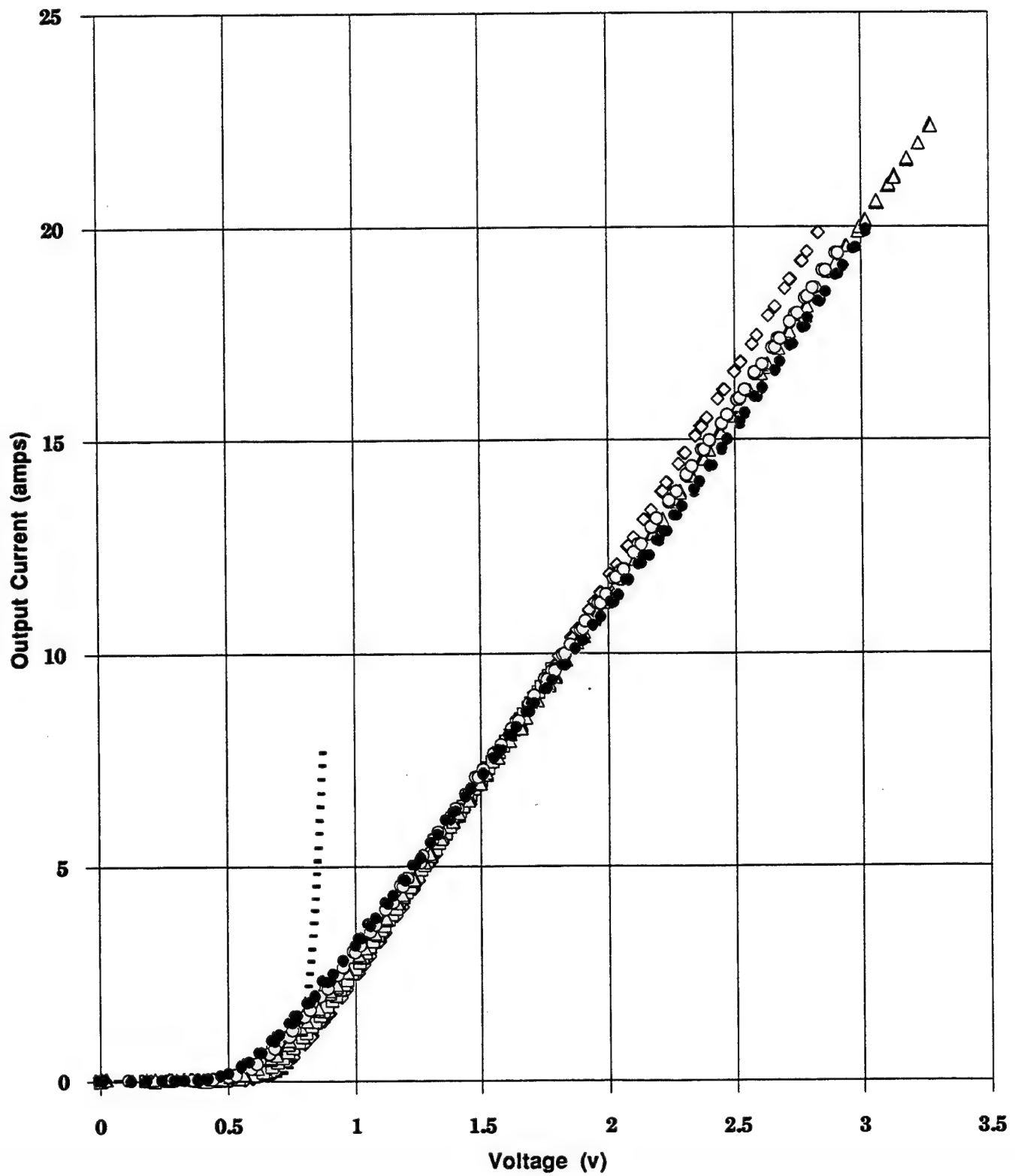


Fig. 4
37-11

display this characteristic, it exists as a trend for this collection of data. From 0 volts until about -1100 volts, aside from the 75C data, for a given voltage the absolute value of the current is larger for higher temperatures. This is because as the absolute value of the voltage, or the potential difference across the depletion region between the n- and p-type materials increases so does the energy barrier. For higher temperatures, the electrons have more energy, allowing more of them to cross the energy barrier, which in turn increases the absolute value of the current. Since the depletion region widens in the reverse bias, before breakdown, tunneling is not an option and the few electrons that cross the barrier do so with the aid of outside energy such as light, heat, or collisions. This flow of electrons takes the form of the reverse saturation current. With an increase in temperature the reverse saturation current is greater at a given voltage prior to breakdown. This is due to the fact that heat provides electrons with additional energy. It is this extra energy that propels the electrons over the energy barrier. Eventually breakdown occurs, when large amounts of voltage are being applied and a vast number of electrons are then able to travel through the barrier, rather than having to jump over it.

After -1100 volts, a new trend occurs. At this point the graphs show that for a constant current, lower temperatures require less voltage to attain the same value for current. See Fig. 5 for more detail. This goes back to the vibration patterns. For higher temperatures, the magnitude of vibration is greater. This makes it more difficult for the electrons to pass through

the region, and so more voltage is required to attain the same amount of output current. The characteristics of forward and reverse bias are shown together in Fig. 6 (note the change in scales along the axis).

A further analysis was performed on this diode. It was found, that through solid-state physics, the diode current can be mathematically related to the temperature (T) and applied bias (V), so that $I = I_0[(e^{-qV/kT}) - 1]$, where I_0 is the reverse saturation current, k is a constant, and q is the electric charge. By manipulating this equation, $\ln(I) = \ln(I_0) - (qV/kT)$ was derived. From this the relationship between $\ln I$ and $1/T$ was explored. The graph in Fig. 7 shows that they are related through a curve. In this case, in order to achieve a value for the reverse saturation current (y-intercept), a curve was fitted to the original data points for each temperature near .67 volts. The curve was then extended and the reverse saturation current was found to be 2.225625 Amps. The output current varies significantly with each temperature increment. This indicates the magnitude of the impact that differences in temperature can have on a diode.

A weak relationship exists between the case temperature and the lead temperature. As the case temperature increases to attain specified temperatures, the lead temperature increases as well, but always remains slightly lower than the case temperature (refer back to Table 2). This is probably due to the setup where the lead is further removed from the heater stage than the case (Fig. 2). The case is resting directly on top of the heater stage and gaining heat through conduction. The leads,

DIODE REVERSE CHARACTERISTICS

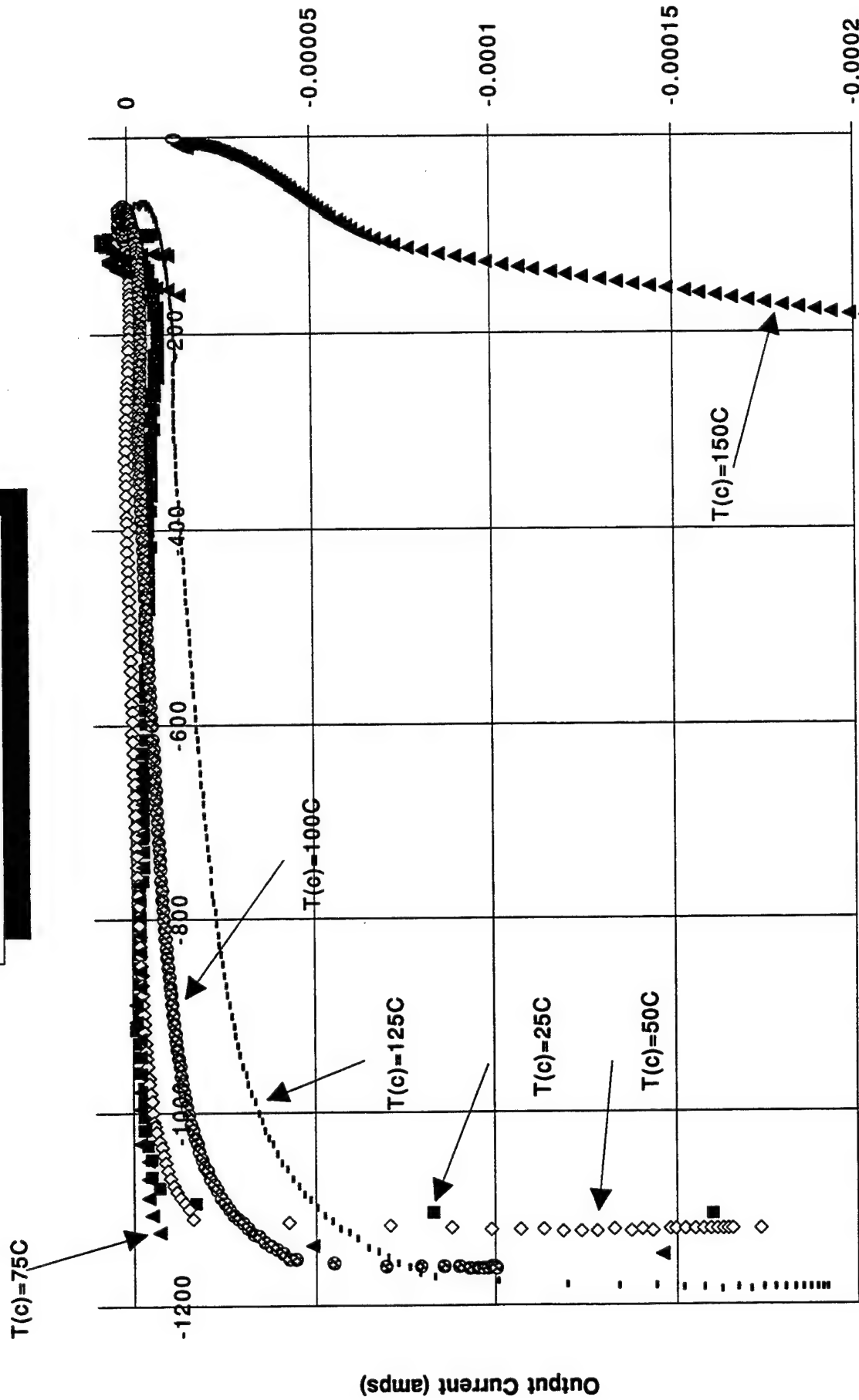


Fig. 5

DIODE CHARACTERISTICS

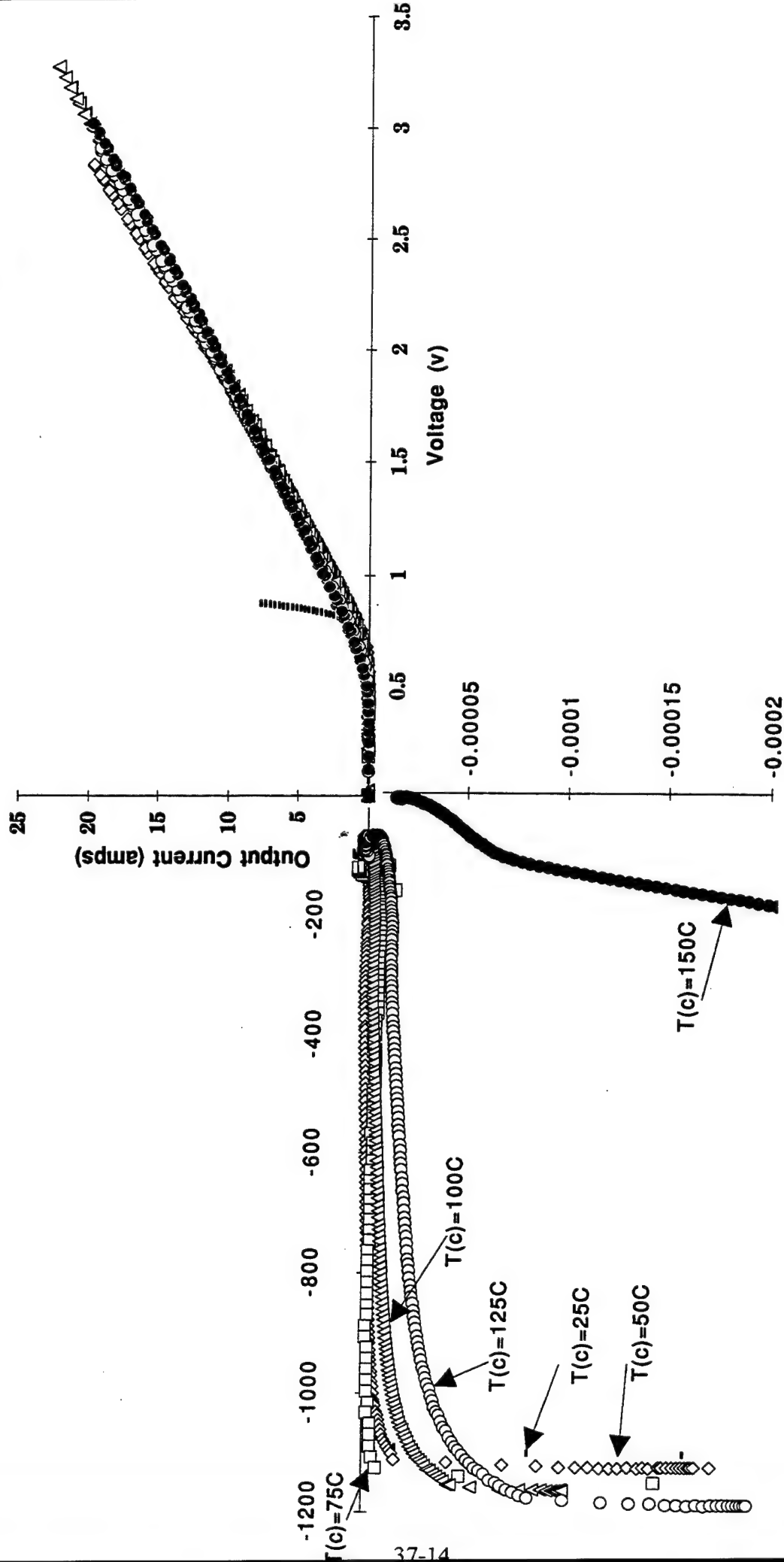
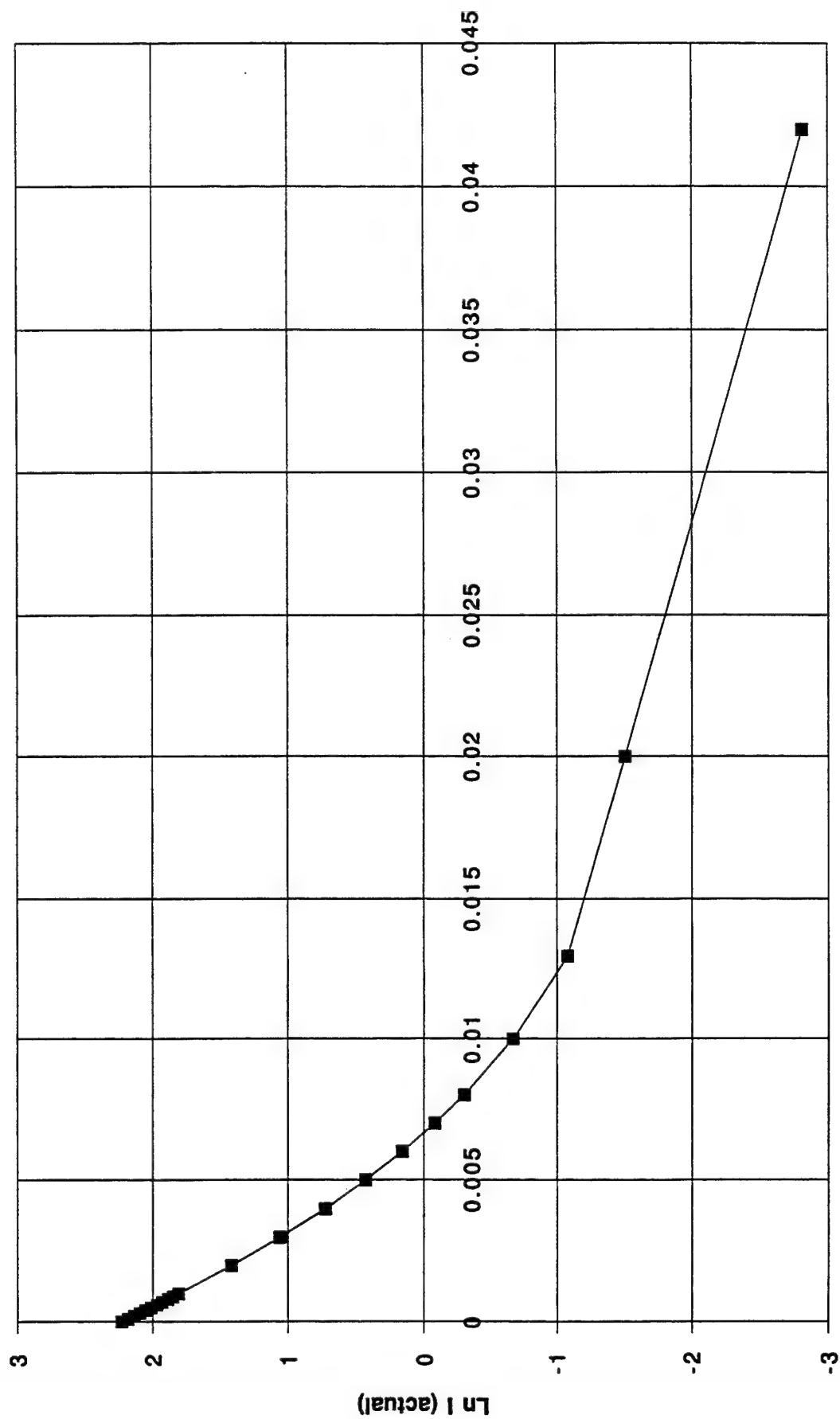


Fig. 6

Diode Forward Characteristics of Inverse Proportionality



1/T

Fig. 7

however, are acquiring heat through the process of conduction, from the heater stage through the case and into the lead; radiation, directly from the heater stage to the lead; and due to internal heat generated by the current flow. The temperature differences between the case and the leads are a direct result of the process of conduction transferring heat more rapidly than the process of conduction, radiation, and internal heat generation, for the temperatures tested in this experiment.

CONCLUSIONS

In this experiment, the characteristics of a silicon diode in forward and reverse bias were tested at different temperatures. The temperature made a definite difference in the performance of the diode. For the forward bias, at small positive voltages, the higher temperature settings produced more output current than the lower temperature settings at the same voltage, up to the point of conduction. At larger voltages (and higher temperatures) the opposite is true, because of increased internal resistance. In the reverse bias condition, the greater temperature settings produce a greater absolute value of output current than the lower temperature settings for the same relatively low voltage (before breakdown occurs). After breakdown, the lower temperature readings show that they require a smaller absolute value of voltage in order to attain the same output current value as the higher temperatures. The reverse conduction voltage gradually increases with an increase in temperature, thereby

contributing to inefficient switching as a result of an increase in lag times due to increased internal resistance. With a rise in temperature, more electrons break their covalent bonds because of absorbed thermal energy. This allows more electrons to cross over the energy barrier. For this reason, at higher temperatures, more current is produced for a given voltage. This remains true until the diode reaches breakdown (in the reverse bias) or the region of conduction (in the forward bias).

REFERENCES

- John P. McKelvey, Solid State Physics for Engineering and Materials Science, Kreiger Publishing Company, Malabar, FL, C.R. 1993.
- S.M. SZE, Semiconductor Devices Physics and Technology, AT&T Bell Laboratories, Murray Hill, NJ, C.R. 1985.
- C.A. Neugebauer, A.F. Yerman, R.O. Carlson, J.F. Burgess, H.F. Webster, and H.H. Glascock, The Packaging of Power Semiconductor Devices, Gordon and Breach Science Publishers, New York, NY, C.R. 1986.
- Robert Boylestad and Louis Nashelsky, Electronic Devices and Circuit Theory, Prentice-Hall, Inc., Englewood Cliffs, NJ, C.R. 1982.

S.M. SZE, Physics of Semiconductor

Devices, John Wiley & Sons, Inc.,
New York, NY, C.R. 1969.

Motorola Rectifiers and Zener Diodes Data Book,

Motorola Inc., C.R. 1988.

DEVELOPMENT AND TESTING OF AN AUTOTUNING CONTROLLER
FOR A TITANIUM:SAPPHIRE LASER

Franklin K. Reyher III
WL/MNGS

Niceville Senior High School
800 E. John Sims Parkway
Niceville, FL 32578

Final Report for:
High School Apprenticeship Program
Wright Laboratories

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratories
Eglin Air Force Base, FL

August 1995

DEVELOPMENT AND TESTING OF AN AUTOTUNING CONTROLLER
FOR A TITANIUM:SAPPHIRE LASER

Franklin K. Reyher III
WL/MNGS
Wright Laboratories
Eglin Air Force Base, FL

Abstract

In experiments that consist of taking data at various wavelengths using a Titanium:Sapphire laser over a large wavelength range at a small step size, there is often such an extensive amount of data to be read that the acquisition process takes several hours. Therefore, it is desirable to automate the process of data collection, allowing the researcher to spend his time more efficiently while simultaneously speeding up the process of data acquisition. An autotuning controller for a Titanium:Sapphire laser, or Ti:Saph, was developed using existing laboratory instruments, namely a New Focus Micrometer Replacement Actuator (MRA) and a Burleigh WAVEMETERjr. These instruments were controlled via LabVIEW on an Austin 486/66 computer by a Lab-PC+ card and the serial port. After the autotuner was designed and worked robustly, a simple absorption experiment was carried out to test the efficiency of the autotuning controller.

DEVELOPMENT AND TESTING OF AN AUTOTUNING CONTROLLER FOR A TITANIUM:SAPPHIRE LASER

Franklin K. Reyher III

Table of Contents

1. Introduction
2. Methodology
 - 2.1. The MRA
 - 2.2. The WAVEMETERjr
 - 2.3. The Entire System
 - 2.4. Testing the System
3. Application
4. Results and Conclusions
5. Recommendations
6. References
7. Figures

DEVELOPMENT AND TESTING OF AN AUTOTUNING CONTROLLER FOR A TITANIUM:SAPPHIRE LASER

Franklin K. Reyher III

Introduction

In the typical laboratory, experiments are usually performed by researchers using an instrument or two to aid in the acquisition of data. If there were some way for the experimenter to perform other necessary tasks by automating the actual data collection process, the researcher could raise his level of efficiency while expediting the research. Many laser experiments are comprised of measuring certain physical properties over a certain range of wavelengths. Some lasers, the Titanium:Sapphire (Ti:Saph) being one of them, can vary in wavelength by way of a rotating a birefringence tuner, which is commonly called "tuning" the laser. At present, the researcher is forced to tune the laser to the desired wavelength and hand-record data over a range that may be as great as one hundred nanometers at a step size of one-half nanometer -- two hundred data points. When data acquisition takes hours upon end, a new system must be found to automate and replace the process of human operation, allowing the researcher to do other necessary laboratory tasks while simultaneously speeding up the process of data acquisition. Such systems are available at the present, but only if the researcher is willing to set aside upwards of \$30,000 for an automated research system and devote time to customization before use in any case. In order to avoid this high-priced alternative, an autotuning controller for a Ti:Saph laser was developed using existing laboratory instruments.

Methodology

The two main instruments that were used in the autotuner's development were a New Focus Micrometer Replacement Actuator (MRA) and a Burleigh WAVEMETERjr. These two instruments were controlled with the object-oriented graphical programming interface, LabVIEW, on an Austin 486/DX2 66 MHz PC. The LabVIEW programs (called Virtual Instruments or VIs[1]) that control these instruments were developed over the course of several weeks with much emphasis placed on the system's robustness and the instruments' ability to cooperate with each other and the system. Therefore, it is best to focus on the development of the programs that control these instruments.

- The MRA

The main component of the New Focus Micrometer Replacement Actuator is the Picomotor[2]. The Picomotor works by causing a screw to rotate between two jaws in much the same way as the screw would rotate if it were held between two fingers sliding in opposite directions. A piezoelectric transducer slides the jaws, and the voltage sent to this piezo determines the speed of the screw's rotation. The screw rotates if the Picomotor acts slowly, but fast action causes no rotation due to the screw's inertia.

The MRA was controlled with the Austin's Lab-PC+ board. This board allows for two modes of control - analog and digital. The first method used was digital, because it was thought that digital control would be faster and more efficient. The MRA has two digital control lines, STPAB and DIRAB. STPAB is the step command, and a falling edge of a pulse causes the MRA to step once. DIRAB is the direction input; a high level (+5 V) causes clockwise stepping, and a low level causes counterclockwise stepping. The program was able to send a pulse signal to the MRA but the instrument would simply not move. To see if the VI was working correctly, the pulse from the Lab-PC+ board was measured with an oscilloscope, and matched the proper output. The problem was eventually discovered - the MRA wouldn't run because it had a blown fuse. This final obstinate component was replaced, and the screw began to turn. Very slowly, too, because the VI sent the MRA a maximum step of 2 mrad at 10 Hz, and individual stepping wasn't near fast enough. For testing purposes, the MRA had been connected to an electro-optic stage, and the change in the stage's position was barely noticeable after ten minutes of MRA operation. Surely there was a better way, and the ultimatum was given: control the MRA in analog mode or simply write the program in C. The former was chosen, as the latter would surely have been much more painful. It was a great surprise that analog control was actually much faster than digital control. With analog control, the step rate is actually variable depending on the voltage sent through the line APOT (which replaces STPAB) to the MRA. This step rate increases exponentially at first and then linearly. The new analog program, Analog Drive.vi, pushed the MRA to 5 5/8 RPM, a faster rate than the published top speed. With analog control, it was not necessary to input a pulse wave to the MRA - voltage data ranging from -2.4V to 2.4V could simply be used. A positive voltage moved the MRA counterclockwise and subsequently up, while a negative voltage moved the MRA down by turning clockwise. Analog Drive.vi was a very user-friendly program, allowing data to be input describing direction of rotation and amount to run. The amount to run specified was a function of how many times to run a certain for loop. When the program eventually evolved into Analog Motor Driver.vi, this for loop was replaced by a timer delay and the program became simplified. The fifth version, or Analog Motor Driver 5.vi contains the following (see Figure 1): Wavelength, or the actual wavelength of the laser light, and Desired Wavelength, simply the wavelength needed for the particular experiment. Wait time is a calculated number of milliseconds to wait based on the distance between Wavelength and Desired Wavelength; Length to Travel can be set either to Full Length, 7/8 Length, 3/4 Length, and Half Length. The purpose of the Length to Travel variable is to take the laser to the desired wavelength as quickly as possible while not exceeding the destination and causing hysteresis; this will be discussed in depth later. Device and Channels (0) are parameters for the interface card connected to the Lab-PC+ board. Voltage is simply the variable amount of voltage sent to the MRA - it is usually set to the maximum, 2.4V. Finally, Direction has two settings - if set to counterclockwise, no change is made in the program; if set to clockwise, the Voltage becomes

negative. With analog control, the MRA was now working correctly, properly, quickly, and perhaps as efficiently as possible.

- The WAVEMETERjr

The Burleigh WAVEMETERjr was the next instrument to be brought under control. Control, however, is not exactly the correct word, since it is impossible to write commands to the WAVEMETERjr through the RS-232 interface. The WAVEMETERjr is a “talk only” device, and data could only be read[3]. This data was transmitted at 2400 baud in the form “wavelength, unit measure, atmosphere.” An example is “797.10 NANOMETERS AIR;” there are two possible unit measures (nanometers or inverse centimeters), and two different atmospheres (air or vacuum). The VI for Read Wavelength 5.vi is quite a bit simpler than the MRA driver; first it gives the PC the parameters required to locate the WAVEMETERjr, then the VI displays current wavelength and a graph to display the tracking of wavelengths (see Figure 2). Read Wavelength 5.vi only reads one wavelength, but the data accumulates on the graph if it is not cleared. The major problem that was encountered when writing the VI that displayed the WAVEMETERjr’s output was the way that the instrument sent data to the PC. This data came into the serial port as a string consisting of the wavelength, units, and atmosphere, followed by a carriage return and a line feed. The PC starts reading the string when the user executes the program, so these reading usually started in the middle of the string. In order to position the wavelength data first, the VI searched for a sequential carriage return and line feed. After these closing parts of the string were located, the PC could begin to read the data starting from the beginning of the string. This wavelength data can be taken out of string format and converted into a numeral. This numeric data is then inserted in the graph, and can be used later. If the WAVEMETERjr sends a LO SIG (signal intensity too low), a HI SIG (signal intensity too high), or a COH ER (coherence error) to the PC, Read Wavelength 5.vi will produce a zero as the numeral and will cause the system to beep, alerting the user that an error has occurred.

- The Entire System

Once the MRA and the WAVEMETERjr worked cooperatively as separate components, it was time to unify their operation. This was done with Autotuning Controller.vi, which went through several various versions and in the end became Autotuning Controller 5.vi. This VI is largely just a combination of the previous two VIs (see Figure 3). Port, Device, and Channels (0) are the same parameters that were sent to the MRA and WAVEMETERjr. Desired Wavelength and Length to Travel were also sent to the MRA in Analog Motor Driver 5.vi. The two panels on the lower left, however, are new to the autotuner program. These are error-correcting measures, both designed to overcome laser instability. These will be discussed in the next section along with a full explanation of the Length to Travel variable. The program’s execution is simple; first, it reads the wavelength and averages, taking the median wavelength value. The difference between this median wavelength (Wavelength) and Desired Wavelength is calculated, and if the difference is greater than 0.1 nm of each other, the required commands are sent to Analog Motor Driver 5.vi. Voltage is input to the MRA, and it runs for a certain number of milliseconds before stopping

(calculated in Analog Motor Driver 5.vi by multiplying the difference in wavelength by 8000 ms, a value discovered after much experimental testing). As the autotuner takes data over several wavelengths, this data is output from Read Wavelength 5.vi to an array. When Wavelength and Desired Wavelength come within .1 nm of each other, the program stops, but not before asking the user if he would like to save the autotuner data. This data is written to a spreadsheet file and can be viewed as a graph in any spreadsheet program.

- Testing the System

Once the first two VIs were placed inside a preliminary version of Autotuning Controller 5.vi (which subsequently made them sub VIs), the autotuner was ready to be run. The first test run went well, but the next day the controller didn't work at all. Adjustments were made on the averaging process, and the autotuner worked satisfactorily over the wavelength range set by the birefringement tuner. When the wavelength was pushed beyond the boundaries of the birefringement range, the system crashed, as was to be expected. The most important test to be run, however, was that of the "Length to Travel" experiment. The autotuner was run from a wavelength of 790 nm to 810 nm at each different length: Full, 7/8, 3/4, and Half. Figure 4 displays these tunings superimposed on one another. It is readily apparent that, though both Full and 7/8 take an equal number of steps to reach 810 nm (four), the Full Length Travel overshoots the desired wavelength and then overcorrects. 3/4 Travel takes half again as long to reach 810 nm, and Half moves more than two times as slow. The data indicates that a travel length between Full and 7/8 is optimal, and, for now, 7/8 is the most efficient level the system can be run at, especially because of the undesired hysteresis effect experienced by the autotuner when the wavelength exceeds the desired amount. The next step in the testing process was to add averaging. Figure 5 is largely a graph of laser instability, and you can see that between 850 nm and 852 nm there are numerous unexplained mode shifts. To tune to a correct wavelength, the autotuning controller would have to maneuver around these unstable points. It is apparent that, with averaging on, the controller drives the wavelength down to its desired position even though it passes through several highly unstable regions. Averaging was first done using a mean function; later it was changed to utilize a median function. Autotuning Controller 5.vi also had another preventative measure for laser instability: the Number of Seconds to Wait panel on the VI front panel also helps in an autotune by allowing the laser time to stabilize before taking wavelength data. Finally, with the autotuner working correctly, it was time for an experiment.

Application

Two experiments were conducted with the autotuner, though they were nearly identical. The setup was nearly the same as when the autotuner was being developed as well. An argon gas laser pumped, or supplied power to, the Ti:Saph, the beam of which was then split. Some of this infrared light passed into the WAVEMETERjr and the rest was passed to an open laser cavity. This laser cavity consisted merely of two mirrors and a crystal. The crystal used was the only difference between experiments: first, a

Neodymium:Yttrium Lithium Fluoride (or Nd:YLF) crystal was used, and later a Holmium:Thulium:Lutetium Aluminum Garnet (or Ho:Tm:LuAG) crystal was used. After the beam passed through this cavity, it hit a beam dump. From there, the intensity of the light splashing was measured with a photodiode, or PIN. From intensity data collected prior to passing through the crystal (I_{in}) and afterwards (I_{out}), the absorption coefficient of the crystal can be calculated at a certain wavelength. In order to get a better picture of an absorption curve, it is necessary to collect data over a wide range of wavelengths. To do this, the autotuning controller was used and worked acceptably except, to a small extent, at unstable points. The data collected from the Nd:YLF experiment can be seen in Figure 6, while the absorption curve from the Ho:Tm:LuAG experiment can be seen in Figure 7.

Results and Conclusions

There were four major conclusions reached as a result of the development of the autotuning controller. The first was that analog control of the MRA is ultimately a great deal better than digital control. Even though digital control is much more precise, it is extremely slow, turning the MRA nothing near its analog speed of 5 5/8 RPM. And, under analog control, accuracy can be reached and is not sacrificed in the least, though digital control could yield accuracy down to picometers; however, such accuracy is not necessary. The second finding was that the optimal Length to Travel lies somewhere between Full and 7/8. Thirdly, it was concluded that peak absorption of the Nd:YLF crystal occurs at 797 nm with an absorption of $3.3 \pm 0.3 \text{ cm}^{-1}$, a conclusion within a 10% discrepancy of the published measurement[4]. Finally, it was concluded that the peak absorption of Ho:Tm:LuAG between 770 nm and 830 nm was at 787.5 nm with an absorption of 4.4 cm^{-1} . This measurement, however, could not be compared against previously documented research, simply because this result could not be found. Both of the past two absorption values occurred at crystal orientations of $E||a$ relative to the incident laser beam..

Recommendations

Even though the autotuner worked acceptably in its two experiments, it can be further optimized for use not only with wavelength tuning, but also for positioning electro-optic stages and other various types of experiments. The LabVIEW code could be further minimized, and, with extensive work, an error-correcting method of totally disregarding laser instability could be devised. Other than these changes, the autotuning controller stands in usable form for any sort of autotuning experiment. It has already been used to perform variable wavelength experiments by laboratory staff. The uses and applications of the autotuning controller to facilitate experimentation and data acquisition are limited only by the operator's ideas

References

-
- ¹ National Instruments Corporation. LabVIEW for Windows User Manual. Austin, TX: National Instruments Corp., 1993
- ² New Focus, Incorporated. User's Manual: Picomotor-Driven Products and Drivers. Sunnyvale, CA: New Focus, Inc., 1994
- ³ Burleigh Instruments, Incorporated. Instruction Manual: WAVEMETERjr. Fishers, NY: Burleigh Instruments, Inc., 1993
- ⁴ Zayhowski, J. J., and J. Harrison. "Miniature Solid-State Lasers." To be published.

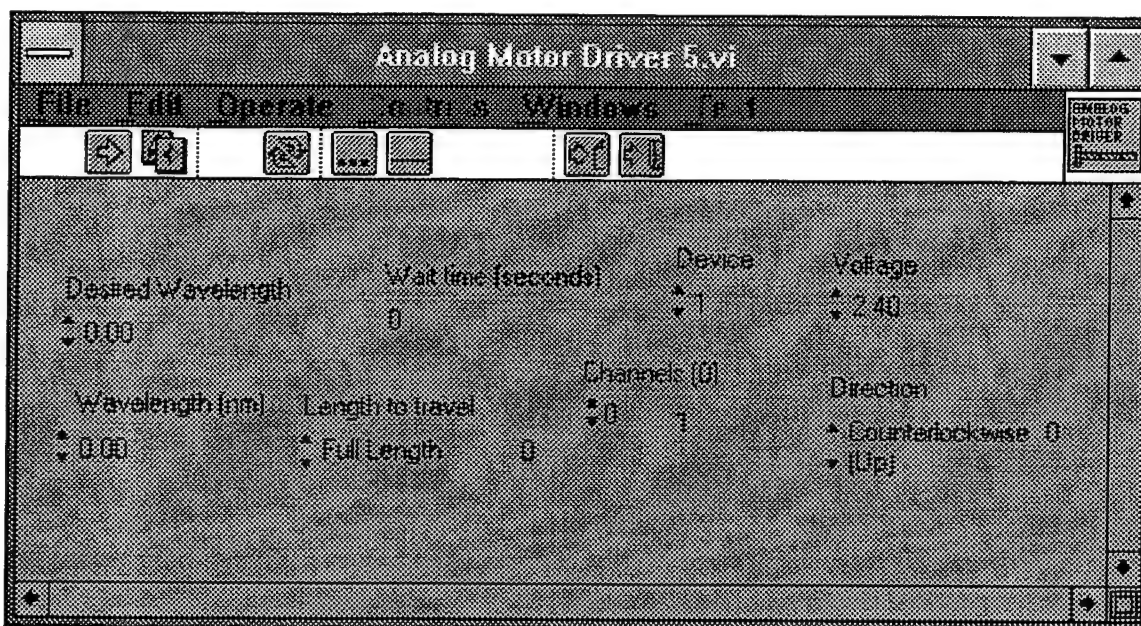


Figure 1: The VI that controls the New Focus MRA.

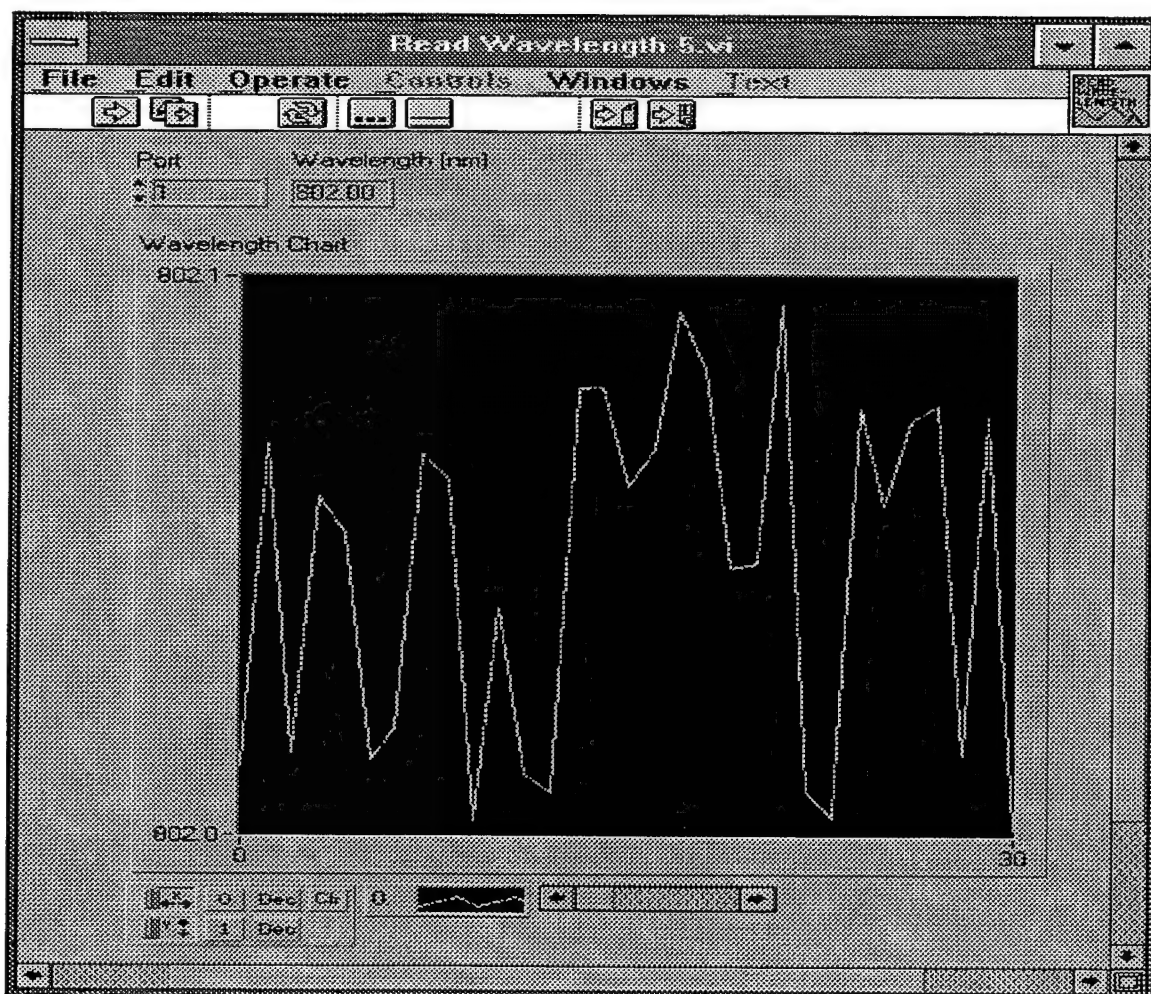


Figure 2: The VI that takes data from the Burleigh WAVEMETERjr.

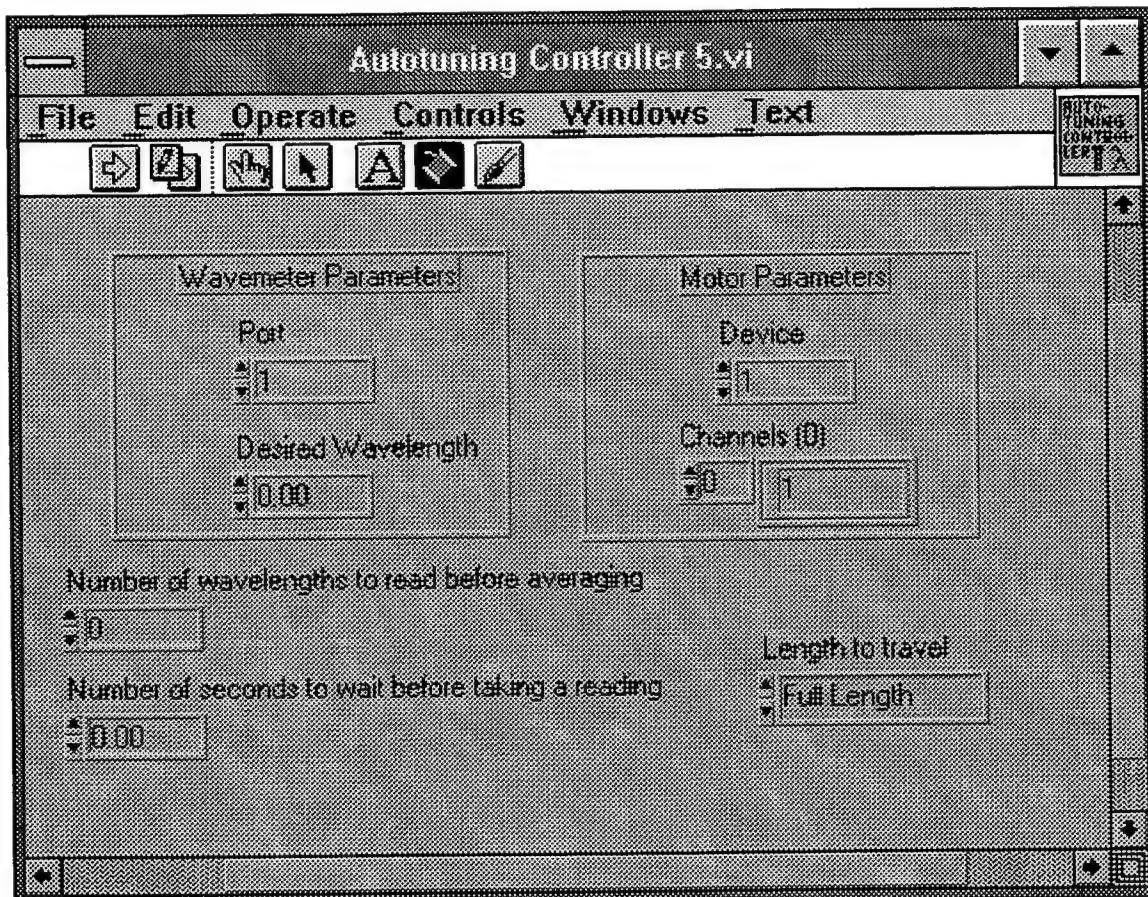


Figure 3: The VI that controls the autotuning controller.

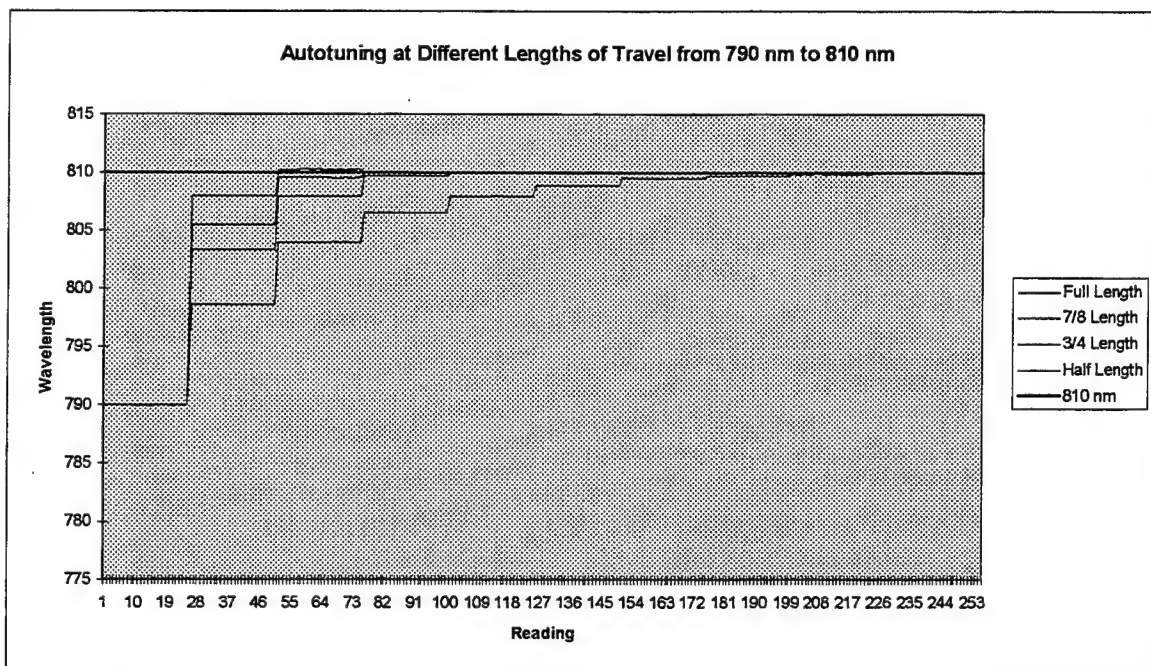


Figure 4

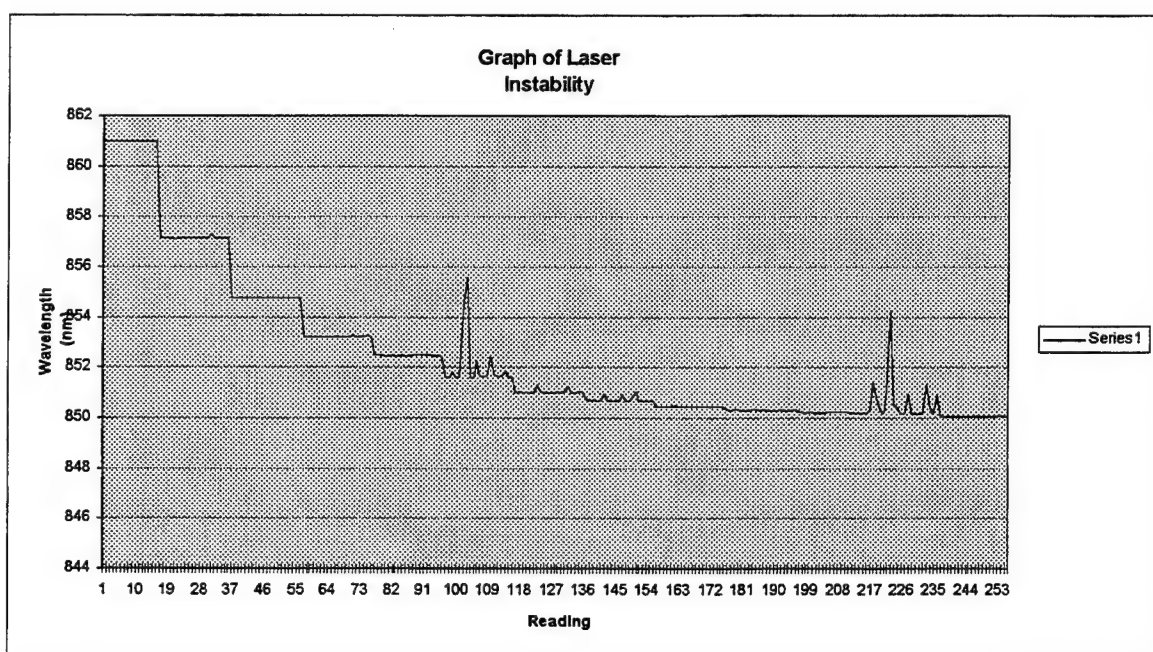


Figure 5

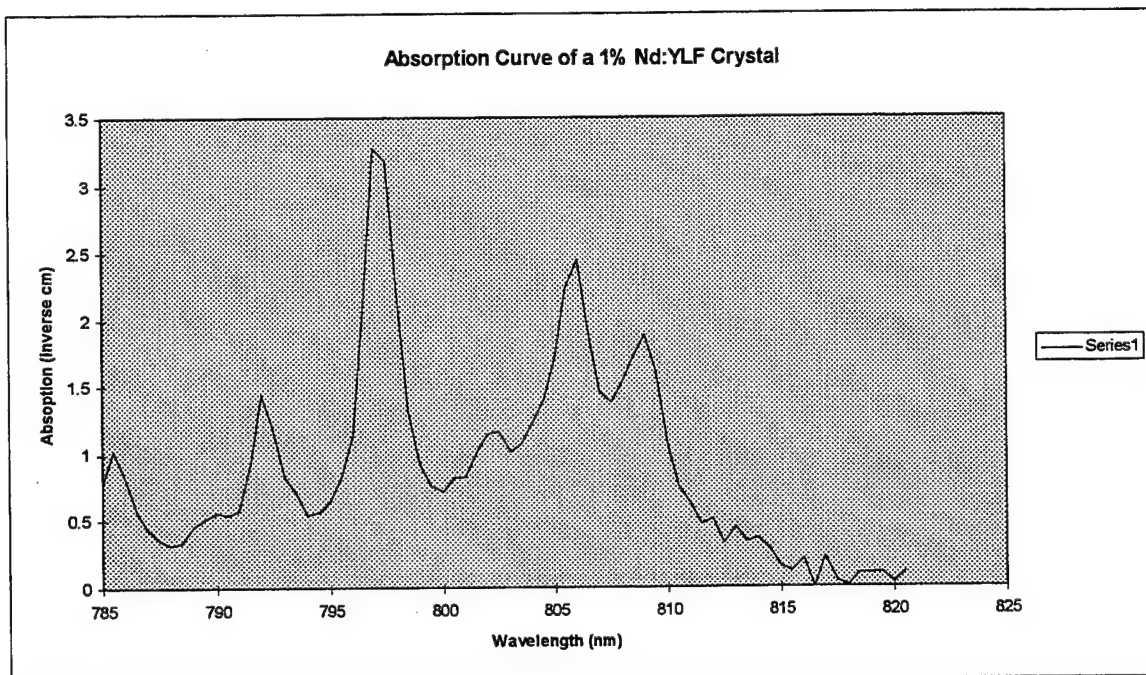


Figure 6

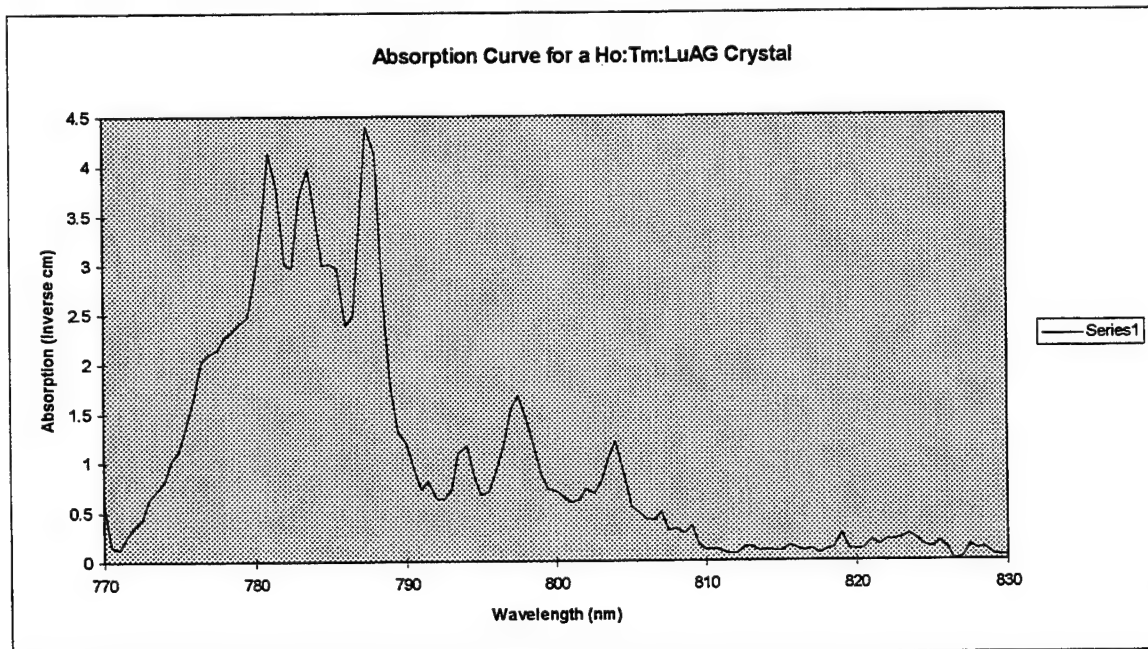


Figure 7

NEURAL NETWORKS AND
DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

Douglas M. Ritchie

Niceville High School
Niceville, FL 32578

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995
39-1

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

Abstract

The topic of interest was Neural Networks, Digital Image Processing, and the possibility of a combination of the two. The Neural Network design and testing along with the Digital Image Processing was accomplished through the use of MathWorks' MatLab. The first part of the project was to become familiar with MatLab's Image Processing Capability by programming a scenario for Point-Source Target Tracking. The second part of the project dealt with edge enhancement and boundary detection. The third section of the project dealt with training a Neural Network to classify patterns based on input vectors. The fourth and final part of the project dealt with combining the edge-enhanced image with an association network to determine whether or not it was a target.

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

Introduction

After making their debut in the late 60's, the popularity of Neural Networks declined sharply until the concept of Artificial Neural Systems almost vanished. Recently, however, these biological models have enjoyed a steady increase in popularity and usage due to the rapidly expanding capabilities of processors. Now, Neural Networks dominate a portion of the research and development at Wright Laboratory and throughout the world. Since the publication of Lippmann's 1987 article which seemed to be a player in the repopularization of the models, new applications such as Optical Character Recognition (OCR) and Machine Intelligence have been employed. It is now feasible to use Neural Networks for on-board processing as devices for target acquisition and recognition.

Section I - Boundary Detection

The Image Edge Quandary

Much of the difficulty in target detection and recognition surrounds the process of determining from an image what is a target and what is not. Many of the problems result when the target exists in high-clutter areas or under extreme camouflage. Despite the new advances in Digital Image technology, an adequate solution to these problems has not yet been found. Some of the best solutions are those that involve edge and boundary detection, which attempt to segment every object in an image and then examines regions of interest, which are determined by outline shape. In order to detect objects from acquired images, an accurate and efficient algorithm for edge detection must be established.

Edge Detection Methodology

The algorithm developed this summer is a pixel difference procedure. All of the test images were found in the MatLab Image Library. The images to be processed were transformed from bitmaps to digital images. The algorithm creates a new, blank image to fit the size of the original image. It then operates on a pixel-to-pixel basis. Each pixel in the new image takes on the value of the difference between the original image's corresponding pixel and the pixel directly

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

to its right. In essence, the new image is a record of the steepness of the contrast in the original image. The new image is then gray-scaled using another prewritten MatLab function.

Edge Detection Results

The results of the Edge Detection algorithm varied. The program succeeded in matching the efficiency of the preprogrammed MatLab edge detector in terms of segmentation and sometimes proved to be better. The aggregate resulting image however contained a higher degree of noise. The first image (fig. 1), which was an indexed picture of two trees by a river, lent itself well to the edge detection process. The outline displays both trees adequately along with the shore-line. In addition to this, the original image had a single leaf floating in the river. The edge detector was successful in capturing this object as well.



Figure 1: MatLab image of trees next to a river. The edge detector was successful in processing this image.

The second image to be processed (fig. 2) was a scanned image of a forest scene. This image unfortunately is very poorly lit. The result of this is that the contrast declined sharply. The largest object in this image is a log at the bottom of the image. The log is difficult to see due to the ineffectiveness of pixel difference in a low-light low-contrast application.

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School



Figure 2: MatLab image of forest scene. Low contrast resulted in a poor representation of edges.

Edge Detection Conclusions:

The edge detection algorithm is very accurate in areas of high contrast. Unfortunately, in areas of low contrast, it becomes completely useless. This problem could be overcome by a contrast enhancement preprocessor such as BCS (Boundary Contour System). The results of the processor when applied to figure 1 were acceptable, but when applied to figure 2, the conclusion was disappointing. Another observation is that the program could be deemed inefficient simply on the merit of its slow run-time. Perhaps when compiled it will run more quickly, but in all likelihood it will still be time consuming.

Section II - IRIS Data

The IRIS Classification Problem

According to CLUSPAK User's Guide, "IRIS is a perennial favorite for testing clustering and classification algorithms." The IRIS Classification Problem is based on IRIS data which was collected to be used as classification input vectors for Neural Networks. There are 150 samples taken from three different flowers, the Sestosa(A), the Versicolor(B), and the Virginica(C). Each sample includes the flower's sepal length, sepal width, petal length, and petal width. The purpose for this experiment is to design and train an Artificial Neural System to correctly classify each of the three types of flowers. Three different nets would be modeled and their results would be compared.

IRIS Classification Methodology

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

To solve the IRIS problem, I uses three methods of Backpropagation: Standard Backpropagation, which is the most general Backpropagation algorithm, Improved Backpropagation, which features a momentum constant and an adaptive learning rate, and the Levenburg-Marquardt optimization technique, which also features an adaptive learning rate as well as approximation and optimization theory. Backpropagation is an algorithm designed by Paul Werbos that uses the generalized delta rule. The name "Backpropagation" is used because the network backpropagates the error of the network through all of the hidden layers. The Artificial Neural System was designed with four input vectors. Each input vector is linked directly to each of the eight hidden layer neurons. The eight node hidden layer uses the log-sigmoid(sigmoid) transfer function. Each of the nodes in the hidden layer is connected to the three node output layer which also implements the log-sigmoid transfer function. The only difference in the three tests was the algorithm used to adapt the weights.

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

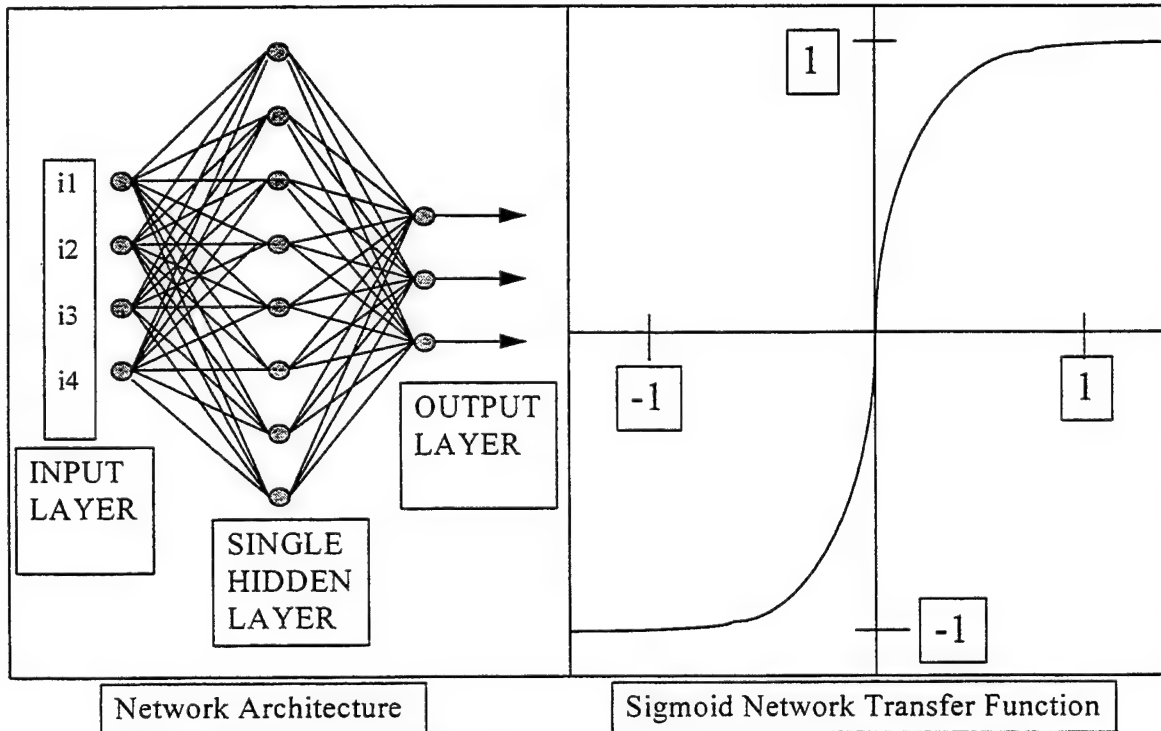


Figure 3a and 3b: Figure 3a(left) is a graphic model of the backpropagation network used. The inputs are labeled i1 through i4. Figure 3b(right) is the transfer function used by each of the nodes in the hidden layer and the output layer.

Standard Backpropagation Algorithm:

$$\Delta W = lr \cdot Dp^T$$

$$\Delta b = lr \cdot D$$

Improved Backpropagation Algorithm:

$$\Delta W(i,j) = mc\Delta W(i,j) + (1-mc)lr \cdot d(i)p(j)$$

Levenburg-Marquardt Algorithm:

$$\Delta W = (J^T J + \mu I)^{-1} J^T e$$

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

IRIS Classification Results

Trial #	Iterations	Time	SSE	Errors
Trial #1	5000	4:30	4.26144	1 B as C
Trial #2	5000	4:29	3.99	1 B as C
Trial #3	5000	4:20	4.14249	1 B as C

Figure 3: Table of results for
Standard Backpropagation

Trial #	Iterations	Time	SSE	Errors
Trial #1	5000	12:34	2.6966	1 B as C
Trial #2	5000	5:09	3.75194	2 B as C
Trial #3	5000	5:18	2.59828	1 B as C

*Time is from 486 DX475 Processor. All other times are from Pentium 90 Processor.

Figure 4: Table of results for
Improved Backpropagation

Trial #	Iterations	Time	SSE	Errors
Trial #1	205	9:24	1.49682	1 C as B
Trial #2	242	5:35	1.23411	None
Trial #3	11	0:06	0.849315	1 C as B

*Time is from 486 DX475 Processor. All other times are from Pentium 90 Processor.

Figure 5: Table of results for
Levenburg-Marquardt Backpropagation

As shown by the three figures(3, 4, 5), the algorithms for Backpropagation had varying degrees of success. The Standard Backpropagation algorithm was the least effective method of solving the problem. After three trials of 5000 iterations each, it achieved what became the highest sum-squared error for the experiment. The processing time was reasonable and the absolute error was consistent at classifying one Versicolor as a Virginica. The Improved Backpropagation made apparent the extra computation involved, as the processing time increased by approximately 45 seconds. The sum-squared error dropped, but, as is the case in many Neural Networks, the sum-squared error did not accurately reflect the actual error. In trial #2, the Improved Backpropagation algorithm made two absolute errors. The Levenburg-Marquardt method boasts the best results of the three algorithms. Although the time loosely translates to about five minutes and thirty seconds, the Optimization Backpropagation has a tendency to be erratic in its computation time. The most important result of the third method was

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

in trial #2 where it solved the IRIS classification problem with no errors. This was the first and only time a network solved the problem with no errors during the experiment. One other surprising result was that the Levenburg-Marquardt Backpropagation reached a sum-squared error of approximately 0.8 during one of the training session in only six seconds.

IRIS Classification Conclusions

The conclusions to this experiment are in the form of suggestions for the applications of these types of nets for classification. The Standard Backpropagation algorithm would be useful in a situation where accuracy was important but speed was more important. The Improved Backpropagation showed little absolute advantage over the Standard Backpropagation, but reached a lower sum-squared error. This could be important in larger scale problems where the number of misclassifications is greater. Although no difference showed up in the absolute classification error, the Improved Backpropagation was actually closer to the solution than Standard Backpropagation. This advantage is to be weighed against the added time the improved version requires. Levenburg-Marquardt could be preferable in the instance that accuracy is key. The Optimization method exhibits various time-saving qualities however that may make it preferable to the other two networks in any circumstance.

Section III - Kohonen Pattern Recognition

The Pattern Recognition Problem

The Pattern Recognition problem in Automatic Target Recognition (ATR) is ever present in today's guidance programs. Much of the difficulty in Pattern Recognition is in segmentation, which can usually be solved with an adequate edge-detector. Unfortunately, edge-detection is not tolerant to noise nor is it 'smart' enough to recognize camouflaged objects. This clutter creates boundary images that are inaccurate, thus resulting in an inaccurate recognition, assuming that the algorithm has any recognition at all. The purpose of this section of my experiment is to develop a system that will scan a binary or gray-scaled image resulting from edge detection, and then create regions of interest for target acquisition.

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

Pattern Recognition Methodology

MatLab was once again the environment used to develop the algorithm for solving this problem. A Neural Network approach would be implemented. Instead of Backpropagation, however, a Learning Vector Quantizer was used in an attempt to create a robust template matching scheme. The Learning Vector Quantizer is an unsupervised Kohonen network that learns to associated patterns based on the similarity of their input vectors. Because the only objective was to prove the ability of the network to recognize objects, the actual accuracy of the training data was irrelevant. Since high-resolution graphics for input files were useless, low-resolution images were created as bitmaps.

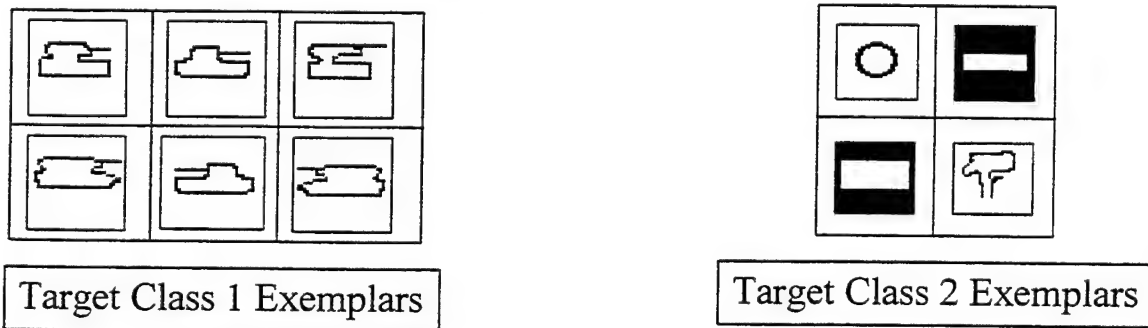


Figure 6: Training images for LVQ network

The images to the left of Figure 6 are some of the target class 1 exemplars used to train the Neural Network. Every iteration meant that the network would get closer to calling images that appear to be similar to these inputs class 1 images. The target class 2 exemplars need not represent anything specific, as long as they were not similar to the tank images.

The second part of preparing the images for training was an attempt to add robustness to the network. The method for doing this was an averaging filter. The hope was that if the network was trained with a blurred image of the original, it would be able to compensate for some of the distortions that are likely to arise in a high clutter environment. Each of the input images were filtered before being entered into the network. Below is a representation of what a filtered image looks like.

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School



Figure 7: Tank image after an averaging filter.

Approximately 15 class 1 exemplars were entered along with 15 class 2 exemplars. After this, the Neural Network was trained with 1024 input characters (each of the pixels in the 32x32 images) which were connected to four competitive neurons. These four hidden-layer neurons were interconnected with four more output neurons that performed linear transfer functions.

Pattern Recognition Results

The results were encouraging from the LVQ network project. The network correctly identified each of the images entered, although no noise was used in the experiment. The network trained on a Pentium 90 processor for approximately 90 minutes. Below are the results of the LVQ Recognition Network.

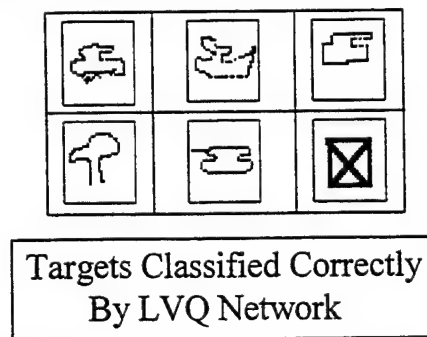


Figure 8: Images correctly classified (From top left to bottom right, tank, tank, tank, not tank, tank, not tank)



Figure 9: Original image before recognition and after(left two), Averaged image before recognition and after(right two)

NEURAL NETWORKS AND DIGITAL IMAGE PROCESSING

Douglas M. Ritchie
Niceville High School

Figure 9 demonstrates the effectiveness of the Averaging Filter technique. The right two images are before the filtering technique was applied. The first image is the input and the second is the networks response to the input, which is a single positive pixel in the center of the recognized area. The second set shows how the recognition broadens with the averaging technique. Training with filters creates a recognition patten almost four times as large, thus making the net more robust.

Pattern Recognition Conclusions

The LVQ network trained relatively quickly and the results were accurate, even more so after the averaging was done. The conclusion is that the LVQ network can recognize targets that it is trained with. This does not, however, mean that this network is applicable quite yet. In order to be useful, the net will have to be trained with a variety of tank images from many different angles. After that, more trials will be necessary to determine whether or not the system can handle noise. The concept of the LVQ network seems to suggest that it will handle noise well, but it will only be able to recognize a region of interest if enough of the actual tank outline is provided.

References

- Lippmann, Richard P., "An Introduction to Computing with Neural Nets," *IEEE ASSP*, 4-13 (April 1987).
- Daniell, Cindy E., "Artificial neural networks for automatic target recognition," *Optical Engineering*, 2521-2531 (Dec. 1992).
- Shirvaikar, Mukul V. and Mohan M. Trivedi, "A Neural Network Filter to Detect Small Targets in High Clutter Backgrounds," *IEEE Transactions on Neural Networks*, 252-257 (Jan. 1995).
- Demuth, Howard and Mark Beale, Neural Network Toolbox, The MathWorks Inc. (1994).
- Bezdek and Pal, CLUSPAK User's Guide, (1993).

Associate did not participate in program.

AN OUTLOOK ON M. B. E. MODELING
USING COMPUTER SIMULATION

Pete Sargent

Wayne High School
5100 Chambersburg Raod
Huber Heights, OH 45424

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling air Force Base, DC

and

Wright Laboratory

August 1995

AN OUTLOOK ON M.B.E. MODELING USING COMPUTER SIMULATION

Pete Sargent
Wayne High School

Abstract

Video animations showing MBE modeling using computer simulations were made. These videos were created for a dual purpose. They were constructed to be used as a reference material in presentations and also as a way of finding optimal growth conditions. All videos show the growths of the super-lattice semiconductor GalliumArsenide (GaAs). By altering the parameters of a program developed by Dr. Don Dorsey (1) and Dr. K. Mahalingham (2), differing simulations were made. These simulations were then put through several computers' graphics programs. This long arduous process culminated into a final video tape, complete with titles, motion, and sound.

AN OUTLOOK ON M.B.E. MODELING USING COMPUTER SIMULATION

Pete Sargent
Wayne High School

Introduction

M.B.E. is an abbreviation for Molecular Beam Epitaxy. It is the latest and most effective process for growing semi-conductor superlattices. Elemental Gallium and Arsenic (other group III-V metals can be used) are vaporized and held in a vacuum at approximately 7.1×10^{-10} torr to prevent them from diffusing. They are then shot in beam form at a zincblende substrate. As the differing atoms hit the substrate they adhere and begin to attract other atoms. They continue in this matter until a super-lattice structure is grown. This process is monitored in two ways, through the use of shutters and temperature. The use of shutters is the primary way of molding the MBE crystals. Each beam is shot from a different source on the vacuum, so that a shutter can be placed in front of either beam and control the flux of both elements. This is useful because it not only allows one to find the most effective growth environment, but also eliminate waste. For example, alone, Arsenic bounces off when shot at a substrate, but when shot in conjunction with Gallium they both stick and begin to grow. Using stoichiometry to determine the mass of the Arsenic, Gallium, and GalliumArsenide particles, a mass spectrometer can be used to accurately determine the amount of each particle being lost. Once this is known the shutters can be adjusted to find the best ration of flux between the two elements.

The use of temperature is becoming obsolete because it requires an extended amount of time to show an appreciable difference.

Methodology

The video animations were made using computer simulations compiled and developed through graphical interfaces. By using a code developed by Dr. Dorsey (1) and Dr. Mahalingham (2) computer simulations of MBE growths were modeled. Certain parameters could be altered to achieve required outcomes. Temperature and substrate size were two of the main stipulations to be dealt with. The substrate tilt was also important and dually affected the substrate size. A special category called "output frequency" was also changed in the code. This was the basis for the production of the animations. As each substrate showed the simulated atom-by-atom pattern a "snapshot" was taken of it and saved as a file. The number of these "snapshots" made was determined by the output frequency. The most common output frequency in our work was twenty, with an occasional case of forty. Now that a series of files had been saved the long laborious task could begin. Each file was stored on a UNIX based Silicon Graphics Computer. The portion of the animation process done here was all through a graphics software program called AVS (Advanced Visual Systems). By creating a network to process each image it was brought onto the screen. Then by using an "animator" module, a keyframe, flipbook style animation could be built. Each snapshot was brought up and stored as a keyframe. This

was repeated until each snapshot in the series was stored. It could now be played back as an animation. The next step was to transfer this animation onto SVHS video. A "video output" module was used from the AVS selection to aid in this procedure. As the animation was played on the computer it was simultaneously recorded onto SVHS video in real time. As this stage was completed the taping half of the project was done. The tape was then recorded onto a Power MacTM computer and put through another software program, "Adobe". Once in the Adobe each animation was saved as a clip. Each clip was recorded at 15 frames/second and the dead frames between each new snapshot were deleted. This reduced the time between new growths and created a smooth transition to the eye. Titles were then added to give background information to the viewer, this was done in conjunction with super-imposed layouts. Next audio was put in where needed. Five keyframe animations underwent this process and were all melded together onto one clip. The whole project was then saved as a QuicktimeTM movie. These movies were then transferred back onto SVHS video in the same manner as the keyframe animations were first done. Once this was accomplished a final video ready for all was completed.

Results

The result of the project was a well-planned and executed video. Researchers will now be able to better assess growth

conditions, and more readily access past cases to find flaws. This will aid in the finding of possible solutions to these problems. It will most of all provide a simple visual medium to help explain the many facets and complexities of the MBE process.

References

1. Dr. Don L. Dorsey Wright Laboratory/MLPO
3500 P Street, WPAFB, OH 45433
2. Dr. K Mahalingham Wright Laboratory/ MLPO
3500 P Street, WPAFB, OH 45433

A Study of 3D Laser Scanning

Kevin C. Schlieper

**Tippecanoe High School
555 North Hyatt St.
Tipp City, OH 45371**

**Final Report for:
High School Apprentice Program
Wright Laboratory**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC**

and

Wright Laboratory

August 1995

A STUDY OF 3D LASER SCANNING

Kevin C. Schlieper
Tippecanoe High School

Abstract

A three dimensional scanning system which utilizes both a camera and laser system was studied. To scan, a high powered laser was spread into a plane which rotated around the target object. Laser line data was recorded with the camera and stored in the personal computer. This data was then processed to present a three dimensional, x,y,z, axis plot.

After improving on the current design, a new, mobile scanning project was activated. This project is still in the preliminary stage, but the plan is to build essentially the same principle as above, but with a few minor changes including the availability of mobility.

A STUDY OF 3D LASER SCANNING

Kevin C. Schlieper

Introduction

The need for different forms of data acquisition for personal computers has grown at a steady rate over the past few years. There have been many forms invented and many of these inventions have been improved upon. This project has taken the principle of two dimensional scanning, which is now readily available to the public, and lifted it to the complex idea of the third dimension. This third dimension would allow real life association with objects. This association allows detailed scrutinization and measurement of objects which might otherwise be complicated or impossible through traditional means. Potential applications include many possibilities in the medical field, such as fitting pilot helmets, prosthesis fitting, and organ scanning. Other forms of applications include military use such as target scanning and entertainment uses for enhanced special effects. This form of data acquisition has a very bright future in many applications.

Problem-1

The initial problem which was to be faced was to acquire a successful scan of a model truck. A few scans, at this point, had been achieved, but none with the accuracy needed to

reproduce an object such as the model truck. The reason for the lack of previous accuracy was due to misalignment of the scanning arc. Absolute precision would be necessary to acquire the data accurately. Along with this, it would be necessary to produce a more accurate range/height map in order for the personal computer to relate each real point in space with the same point in the data.

Methodology

The primary objective was to improve the ability to acquire accurate, clean data. To do this it was first necessary to improve the alignment process. The major need for alignment was due to the pre-constructed arc's lack of accuracy. The arch was not perfectly centered in relation to the center platform where the scan target was to be located. This involved a long process of triangulation between a string dropped plum from the center of the arch to the view of the camera and eventually the location of the laser plane. The string was moved in either the x or y direction in order for it to appear at the same point (pixel value) on the viewing monitor, at any point during its 360 degree rotation. This could eventually be adjusted to the center of the monitor by moving the camera located on the arch. To improve this process a pipe fitting was added to the rough center of the arch, which would allow a piece, intended to center the string directly out of the center of the arch, to screw on. Also, instead of a string, a single thread was used to improve the degree of accuracy.

After accurate camera alignment was achieved it was necessary to align the laser. The laser was also aligned using the process described above. It was necessary for the laser to

be aimed at the same position throughout the entire 360 degree rotation. This process, although time consuming, was not improved upon, with the exception of using a larger piece of translucent tubing for the rough alignment and then switching to a thread for the fine tuning. This process allowed for fairly accurate alignment of the laser plane. Some allowance was made for error due to the width of the laser plane.

With both alignment processes achieved it was possible to scan objects, but not yet possible to view them accurately. In order to do this it would first be necessary to constitute a new range height map. This is what allows the personal computer to compare a certain real life point to that of a virtual point on the monitor, or in the PC's output. The original process of making this map involved holding an object at a certain point and measuring the range from the camera and the height from the floor. This process proved to be extremely tedious, and in no way accurate or practical. Due to this, many new applications were designed and tried. These experiments finally lead to one very simple and expedite process. A program was written which would take 9 staggered points, who's real life positions were fixed and already known, and determine their position in the computer's output. This was a major stepping stone in the advancement of the scanner and the entire process took at most five minutes to complete. Now with this data available the scans could be viewed with increased accuracy.

Results

The results were extremely good. The data which was acquired from the truck was much better than any previous information. The sizing was much more accurate and the detail was good. Some information had been lost at various points about the perimeter of the

truck data. This loss of data can be explained by a problem that wasn't determined until later on in the project. As an object, or parts of an object, move closer or farther from the camera they also move farther from the focus point of the camera, which should be the targets center of mass. This causes these areas to be out of focus along with the laser, which is also out of focus. This can cause distortion of the data and dispersion of the laser's light throughout the lens, so that later in the processing the program which finds the brightest points may discover points that are not accurately represented due to an unfocused image.

Conclusion

This portion of the research was completed much earlier on than what was expected during the tour. Please keep in mind that many other minor projects have been accomplished which have been done for the sole purpose of benefiting the scanner project. This report has been done as an overview to show the successes achieved and the general advancement of the project. It does not, however, show the many trials and experiments leading up to receive the final results.

The project has been extremely successful thus far and will continue long after my departure. I'm very happy to have had this experience with the scanner project and the Air Force in general. I hope that they may benefit from my time at least a fraction of the benefit that they have given to me.

Problem-2

The second problem faced was to begin the development of a portable unit. In the first portion a large fixed arch was used, which held the camera and laser on it, not to mention the calibration aspects would not be variable. In this new project we were to have a unit involving a laser and a camera which were attached to a small helmet. This added many problems to the spectrum of things. The items which became involved were the following: a variable range, height map would be necessary; some form of calibration would be necessary; and light would be a problem.

Methodology

As the idea began to take form, more and more problems arose. It seemed for quite some time to be out of the reach of modern technology, but more and more it is beginning to look possible. The shape the scanner has taken is a very small camera which will be mounted on one side of a bicycle helmet with a small laser plane mounted on the opposite side of the head. This distance will be similar to the distance which exists in the current scanner. The small distance between the camera and laser allows a better definition of range, but if the distance is too large a problem of accuracy can be caused. Data from the camera will be stored in a small notebook computer with a special battery system which will improve its mobility. All data will also be processed with the notebook. That takes care of the system itself, but there is still the complicated process of calibration. In order to solve this problem, a wood model was constructed which would allow the helmet to rest at one end of the model and a white faced plate that could be placed variable distances from the helmet. The idea is to produce a range height map from 128 images for each of

three positions of this board. This would accumulate the needed information for an accurate range height calibration. To solve the problem of light, a special lens which only allows the passage of light that is the frequency of the laser to pass through.

Results

Unfortunately, the only results available at this time are the prototype and the software that has been developed. The program that will take the 128 images and process them has been written and is being manipulated in hopes of speeding it up. The scanner parts are on order and will not be available for experimentation until after I have left the project.

Conclusion

The many contributions to the project here, have been a great success and a wonderful learning experience. The project has been well on its way for some time, that is quite obvious, and it will continue to prosper and grow after I have left. The mobile unit is not at all far from the development of a prototype that I feel will work quite well, due to the experiments of its predecessor.

ADJUSTING THE METHOD USED IN THE DERRINGER
WATERJET IMPACT SENSITIVITY TEST

Ryan C. Sheffield

Niceville High School
800 E. John Sims Parkway
Niceville, FL 32578

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

ADJUSTING THE METHOD USED IN THE DERRINGER WATERJET IMPACT SENSITIVITY TEST

Ryan C. Sheffield
Niceville High School

Abstract

The method used in carrying out the derringer waterjet impact sensitivity test was adjusted in order to make it usable with a new test setup. This test is conducted with a unit known as a water derringer*. During the test, a propellant burns within the derringer, propelling a piston forward, causing water on the opposite side of the piston to be pushed through a nozzle. This nozzle, in turn, projects a high pressure water jet across a containment vessel at the target holder. In actual explosives tests, rather than preliminary tests, an explosive is located in a casting ring centered on the face of the target holder. The water jet is recorded on film with high-speed photography using a Cordin model 114 framing camera with model 1205 turbine, along with two Vivitar 2HV flash units for lighting. Two cubic centimeters (cc's) of WSF8-AA ball powder were used as propellant and a RP-80 detonator was used to ignite the propellant. Various mirror periods and fire pulse delays were tested in order to locate the water jet on film. Once this was accomplished, the derringer setup and test procedure was adjusted in order to obtain the clearest view possible of the water jet.

*See page 43-7 for a graphic representation of the derringer apparatus.

ADJUSTING THE METHOD USED IN THE DERRINGER WATERJET IMPACT SENSITIVITY TEST

Ryan C. Sheffield

Introduction

In the past, munitions have been disposed of by open burning and open detonation (OB/OD). Due to public image of this form of disposal, as well as the environmental concerns it may cause, it has been necessary to find alternative methods of disposing of munitions in an environmentally friendly way. Methods such as steaming and submersing munitions in a hot water bath were developed to remove melt cast explosives, such as TNT, Comp A, and Comp B, from a munition casing. For the newer "insensitive" explosives, such as Plastic Bonded Explosives (PBX's), however, these methods are of little use. Consequently, new methods of removing explosives were developed. One such method is high pressure water washout. During this process, a high pressure (approximately 15,000 psi) water jet is used to "cut" explosive from a munition casing. Due to the large mass and detrimental reaction capabilities of some of the explosives subjected to this process, it is essential to be certain that no reaction will occur as a result of waterjet impact with the explosive. With this in mind, the water derringer was developed for the Naval Weapons Support Center, in order to test the reaction of explosives to a high pressure water jet on a small scale. This test was initially performed at the Rock Mechanics and Explosives Research Center at the University of Missouri-Rolla (UMR).

Discussion of Problem

At the High Explosives Research and Development (HERD) facility, Wright Laboratory, Eglin Air Force Base, FL., a new demilitarization facility is currently under construction. This facility will

house a high pressure water washout system. Before this system is used to remove explosive from munitions, it is important to obtain test results on the sensitivity of the explosives which will be removed. Consequently, a setup is being developed in which the derringer test may be run at the HERD. Although test procedures were developed at UMR to work with their setup, precisely recreating that setup at the HERD posed a problem, mainly due to the difficulty and cost of procuring exactly the same test materials. Accordingly, their test procedures have been used as guidelines while establishing procedures at the HERD, but substantial changes have been made in order to compensate for the difference in equipment.

Methodology

Parts specifications from the Waterjet Impact Sensitivity Manual were followed to recreate the derringer apparatus. (See page 43-7 for a drawing of the apparatus.) WSF8-AA ball powder was used as a propellant, and RP-80 detonators were used to ignite the propellant. The water jet was captured by high speed photography using a Cordin model 114 framing camera with a model 1205 turbine, along with two Vivitar 2HV flash units to provide lighting. The tests were carried out in an explosion chamber and were controlled from a Cordin control panel in a control room adjacent to the chamber. The test was viewed by the camera through a Plexiglas porthole. Six tests were run, with mirror period (MP) and fire pulse delay (FPD) settings based on those from UMR, without locating the water jet. (Mirror period is the time it takes for the mirror in the framing camera to make an entire revolution. Fire pulse delay is the amount of time between the firing of the camera and flashes and the initiation of the firing pulse that causes detonator explosion.) Even with the elaborate camera and lighting system, the jet was not discernible on film. It was hypothesized that this may have resulted from the clearness of the water. Consequently, the next three test runs were carried out with water tinted by methylene blue. The jet was still not seen on the film, so another method was employed to locate the jet. An empty

film canister was placed over the orifice in the head plate of the derringer apparatus, and then the test was carried out in the normal manner. By observing the position and condition of the canister on film, it was possible to center in on the jet by adjusting FPD and MP settings. This procedure was followed twice, in order to provide a reference for choosing camera settings. Using the information obtained from these tests, five more tests were run (without using the film canisters). On the film from the fifth of these tests, the jet was seen.

After locating the water jet, FPD and MP settings were varied in an effort to capture the water jet on film at the best time possible. Once this was accomplished, various factors in the setup, such as flash positions and distance from the porthole to the derringer, were adjusted to improve the clearness of the jet's image on film. In order to accurately determine what was being seen on the black and white negatives that were being used to view the jet, two shots were recorded on color film. This film was then developed into pictures so that the content could be viewed clearly. Judging from these pictures, it appeared that propellant burn products (blow-by) were entering the containment vessel around the piston or cylinder, and may have been obscuring the jet. Accordingly, the pistons and cylinders used in future tests were coated in a heavy vacuum grease in order to seal the gap through which these products could travel.

Another problem, which at first appeared to be blow-by, was later identified as water vapor, based on its image in the color pictures. Since it was continuously emitted from the nozzle ahead of the water jet, it obscured the jet's tip. In an attempt to eliminate this vapor, the containment vessel was purged with helium, in order to dry out moisture that could contribute to the vapor.

Results

Appropriate settings for FPD and MP were found. Vapor began exiting the orifice in the derringer head plate approximately 100 micro-seconds after the charge was fired. The optimal window, which is the time frame it is desirable to have on film, extends to about 350 micro-seconds after

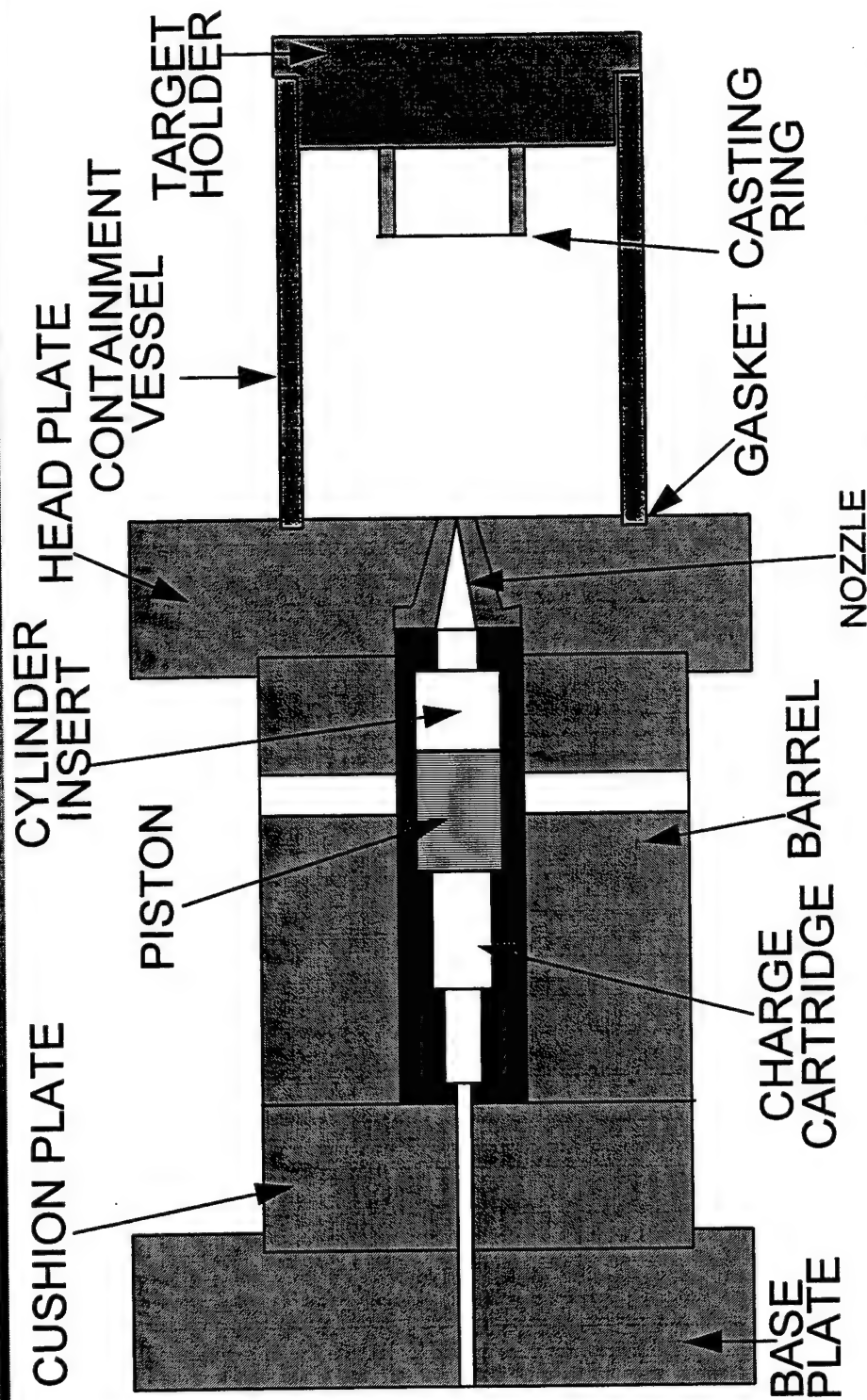
firing time. The optimal mirror period was found to be between 1000 and 1250 micro-seconds. Based on these numbers and the segment of the jet's movement wanted on film, FPD and MP settings can be chosen for each test run. In spite of success in determining these settings, an accurate water jet velocity could not be obtained. The water jet's tip was continuously preceded and obscured by water vapor exiting the nozzle ahead of the jet. Although efforts were made to eliminate this vapor or make calculations around it, none were successful. Since an accurate measurement of water jet movement could not be obtained from negatives or pictures, an accurate water jet velocity was not found. This velocity is needed in order to determine the pressure of the water jet, a key factor in the derringer test.

Conclusion

Since water jet velocity has not yet been determined, the next step will be for Dr. Paul Worsey, one of the supervisors of the derringer tests performed at UMR, to view the test setup and procedure at the HERD and help to eliminate problems. Once this has been accomplished and an accurate velocity measurement can be obtained for the water jet, the test will be run with various explosives in the casting ring, in order to test them for sensitivity. In addition to the procedure followed when running the test previously at the HERD, gas samples will be taken from the containment vessel after the explosive has been impacted by the water jet, in order to detect reactions which produce no visibly noticeable effects.

High Pressure Water Washout System

Impact Sensitivity Testing (Derringer Test)



GRAPHIC COURTESY OF LT. SCOTT JEFFERS

References

Giltner, Scott G., et. al. "Overview of the Demilitarization Program for Class 1.1 Warheads and Rocket Motors Using High Pressure Waterjets at the University of Missouri-Rolla."

Worsey, Paul N. "Waterjet Impact Sensitivity Manual.", Compiled for US Naval Weapons Surface Warfare Center, Crane, Indiana, by Rock Mechanics and Explosive Research Center, University of Missouri-Rolla, in Partial Fulfillment of contract N00164-92-D-0011, 107 pages, December 1993.

A Comparison of Relaxation Spectra From Storage Modulus for AFR700B Advanced
Composite Material

Robert J. Skebo Jr.

Beavercreek High School
2660 Dayton-Xenia Rd.
Dayton, OH 45434

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

A COMPARISON OF RELAXATION SPECTRA
FROM STORAGE MODULUS FOR AFR700B
ADVANCED COMPOSITE MATERIAL

Robert J. Skebo Jr.
Beavercreek High School

Abstract :

The relaxation spectra from storage modulus were investigated for different samples of AFR700B advanced composite material. Baseline values were taken from As-Cured AFR700B and compared to Post-Cured AFR700B which was subjected to an additional high temperature/high pressure autoclave environment. During the experiments, samples of AFR700B were heated to 250°C and 300°C while being physically oscillated every hour for 24 hours. Data compiled from this process was used to graph storage modulus versus frequency and eventually learn more about the composite's relaxation spectrum.

A COMPARISON OF RELAXATION SPECTRA FROM STORAGE MODULUS FOR AFR700B ADVANCED COMPOSITE MATERIAL

Robert J. Skebo Jr.

Introduction :

Advanced composite materials are extremely attractive for integration in fourth generation combat and reconnaissance aircraft such as the B-2, the F-117, the F-22 and other high performance aeriels.¹ Some reasons for the attractiveness of composite materials include a multitude of unique properties like phenomenal stress to weight ratios and low observable properties.

As a result of frictional and explosive forces, engine components on these aircraft heat up to high temperatures (i.e. 700°F). At elevated temperatures, advanced composite material can both chemically and physically degrade. This study focuses on the physical aging aspect of AFR700B with special emphasis placed on the composite's relaxation spectrum .

A relaxation spectrum is a visual display of how an amorphous polymer gradually reaches a state of greater entropy. Organic molecules reach this "lazy" state by unfolding from jumbles into long straight strands. Thermal relaxation phenomenon is like starting with cold spaghetti in a tangle and heating it while pulling it until the spaghetti straightens out. Thus the material (spaghetti) enters a state of increased entropy.

Much like relaxation spectra, every amorphous polymeric system has a glass transition temperature below which the writhing thermal molecular motions essentially cease.³ This temperature is generally not an absolute value, rather it is usually defined as a range. Glass transition is a phase change, and as such this chemical occurrence can also influence the material's mechanical performance.

Glass transition curves are often misinterpreted to be the same as relaxation spectra. Although there definitely appears to be a visual relationship between the two when comparing experimental graphs, conclusions concerning the extent of these mechanical/chemical similarities have not been postulated into definitive theories. This area is currently under investigation by material scientists and engineers.

Objective :

The relaxation spectrum is a visual display of a material's natural tendency to enter a state of entropy. These preliminary experiments are designed to gain a further understanding of relaxation spectra for AFR700B advanced composite material. Samples to be compared include "as-cured" neat resin and "post-cured" neat resin (which was additionally elevated to high temperatures and pressures in an autoclave). This initial study will also evaluate the effectiveness of a data gathering technique using a "Dyna-Stat" frequency modulator and William's & Ferry's³ method of plotting relaxation spectrum from storage modulus.

Methodology :

Materials: AFR700B As-Cured neat resin
 AFR700B Post-Cured neat resin

Specimens: Test specimens will be 40mm x 12mm x 1mm

I. Specimen Preparation

- 1) Cut specimens down to 40mm x 12mm x 1mm with saw
- 2) Sand edges manually to insure easy fit
- 3) Measure exact dimensions and enter into computer
- 4) Set specimen into clamps and tighten

II. "Dyna-Stat" Machine Operation

- 1) Set parameters (i.e. mv/kg, dimensions etc.)
- 2) Close and clamp insulator door
- 3) Set low temp to 249.5 C/high temp to 250.5 C and step up
- 4) Monitor load force and adjust during warm-up

- 5) Lock load adjuster, turn on program, and close loop
- 6) Turn on remote, timer, and printer
- 7) Run for 24 hours (25 total frequency sweeps)
- 8) Turn off remote, timer, and printer
- 9) Unlock load adjuster, turn off program, and open loop
- 10) Set low temperature to 20.0 C and step down
- 11) Monitor load force and adjust during cool-down
- 12) Open insulator door and remove specimen

III. Storage Modulus Graphing Technique

- 1) Copy data for frequency and E' into graphing program
- 2) Change frequency from rad/s to Hz (conversion factor=.159)
- 3) Take the logarithm of frequency in Hz
- 4) For every 6 test hours, plot E' verse log Hz
- 5) Go back to the spreadsheet and add the "A factor"
- 6) Replot graphs to include this time superposition

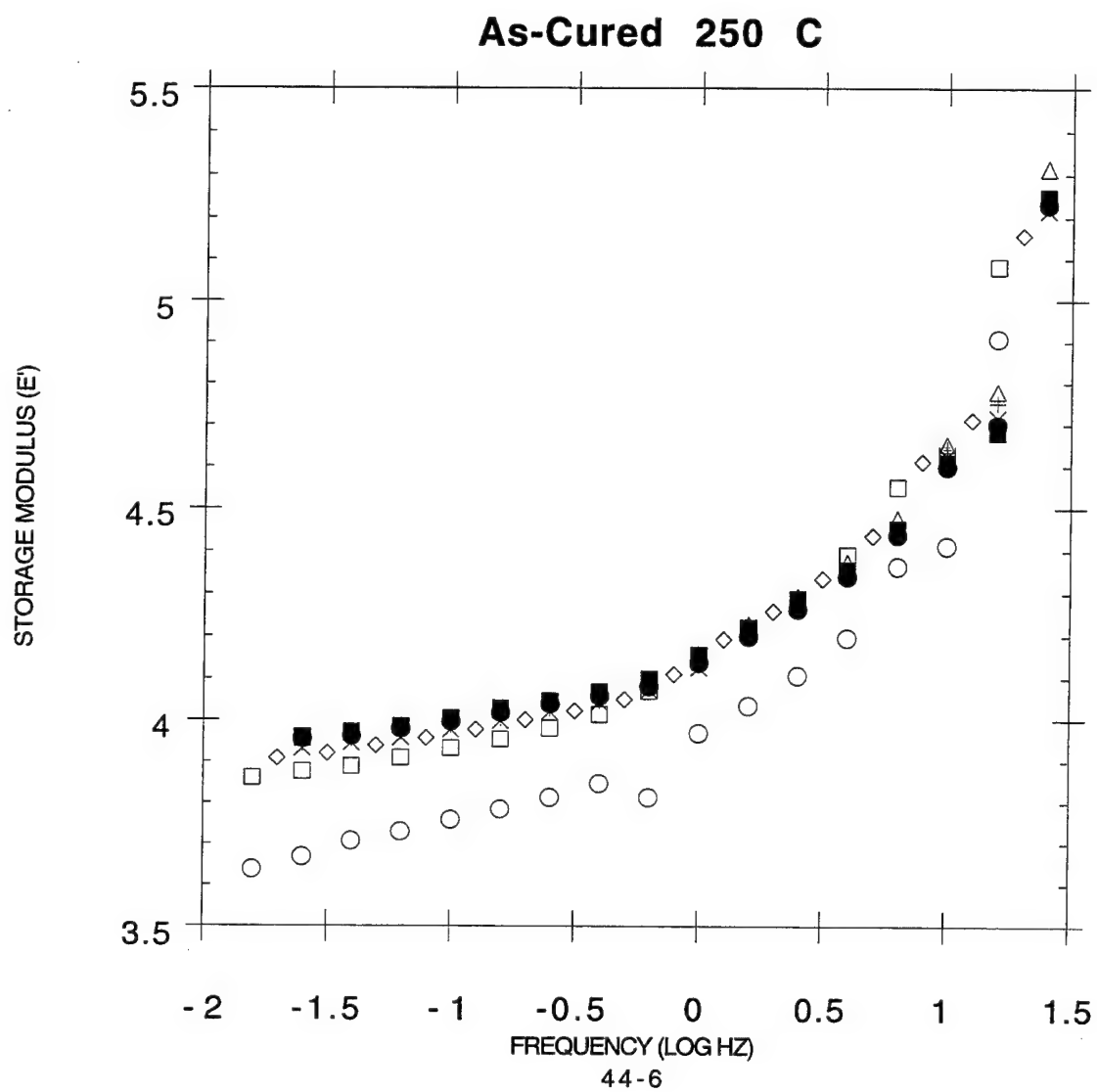
IV. Relaxation Spectrum Graphing Technique

- 1) Copy log frequency and E' data into new spreadsheet
- 2) Use the equation:

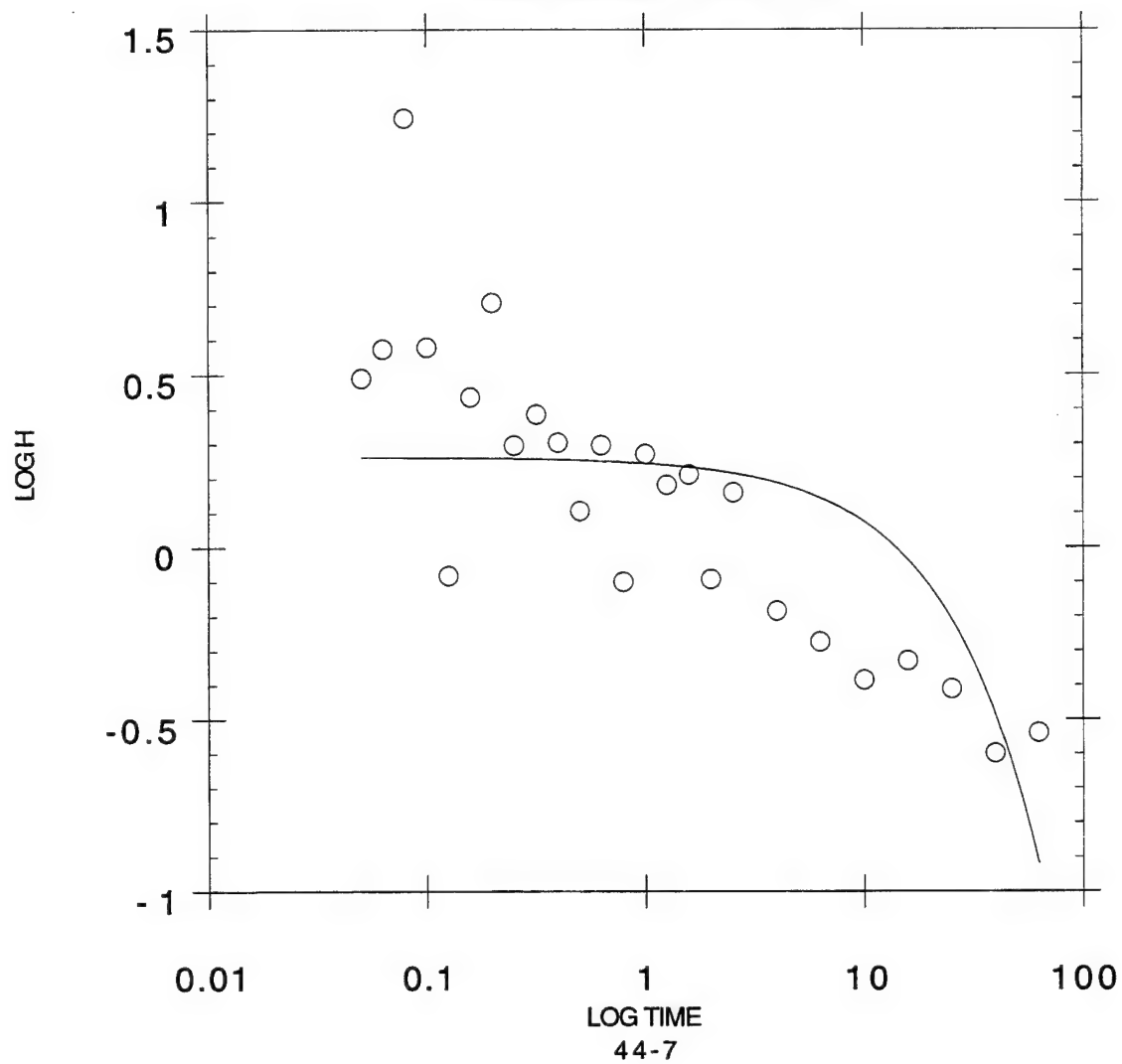
$$H(t)=AG' d \log E'/d \log f$$

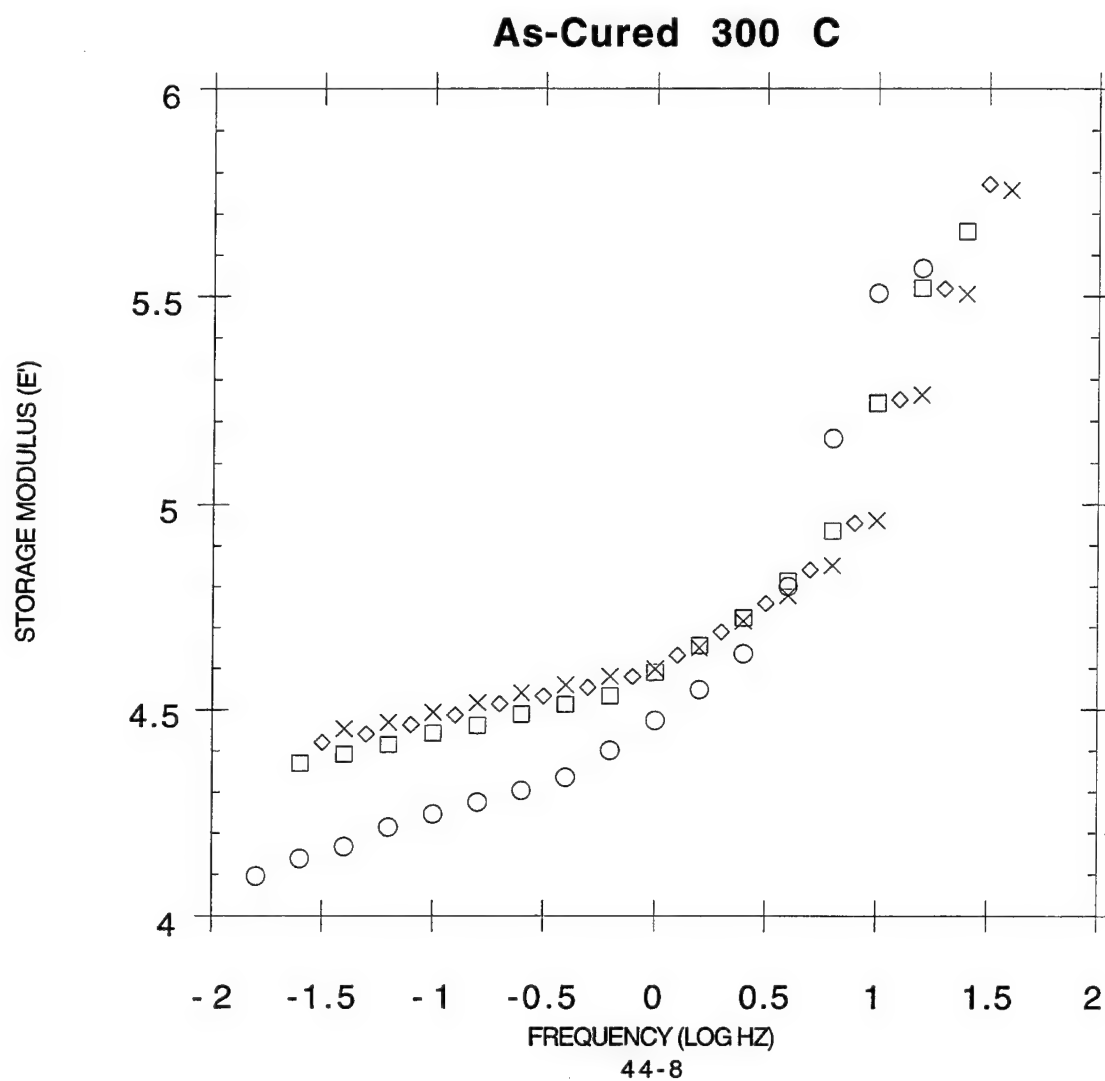
where: H is the positive slope of m from part III,
 A is taken to be 1,
 E' is storage modulus from part III,
 and f is the frequency from part III

- 3) Graph log H verses log time (reciprocal frequency)
- 4) Use the curve fitting option to clarify the graph

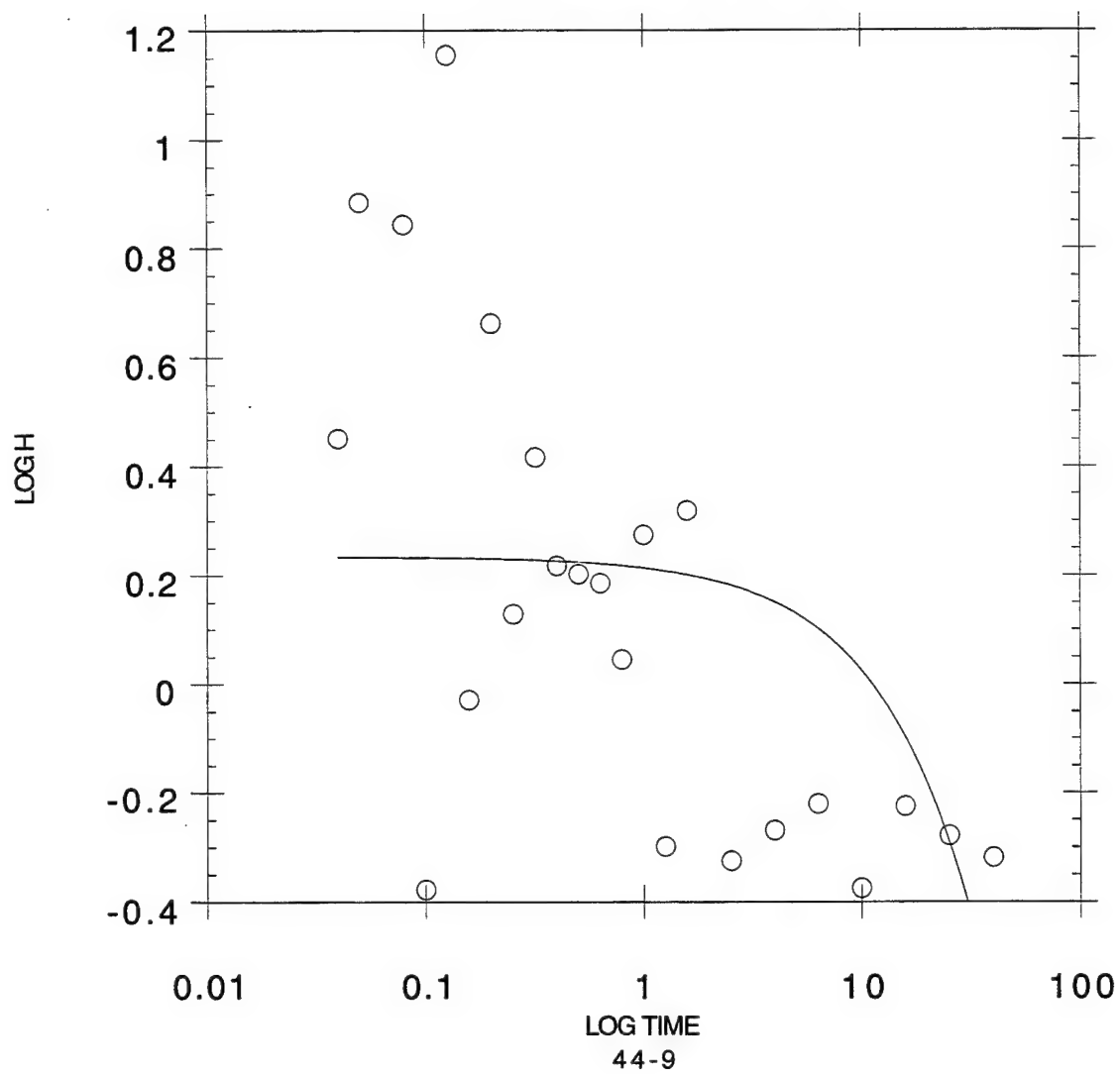


As-Cured 250 C

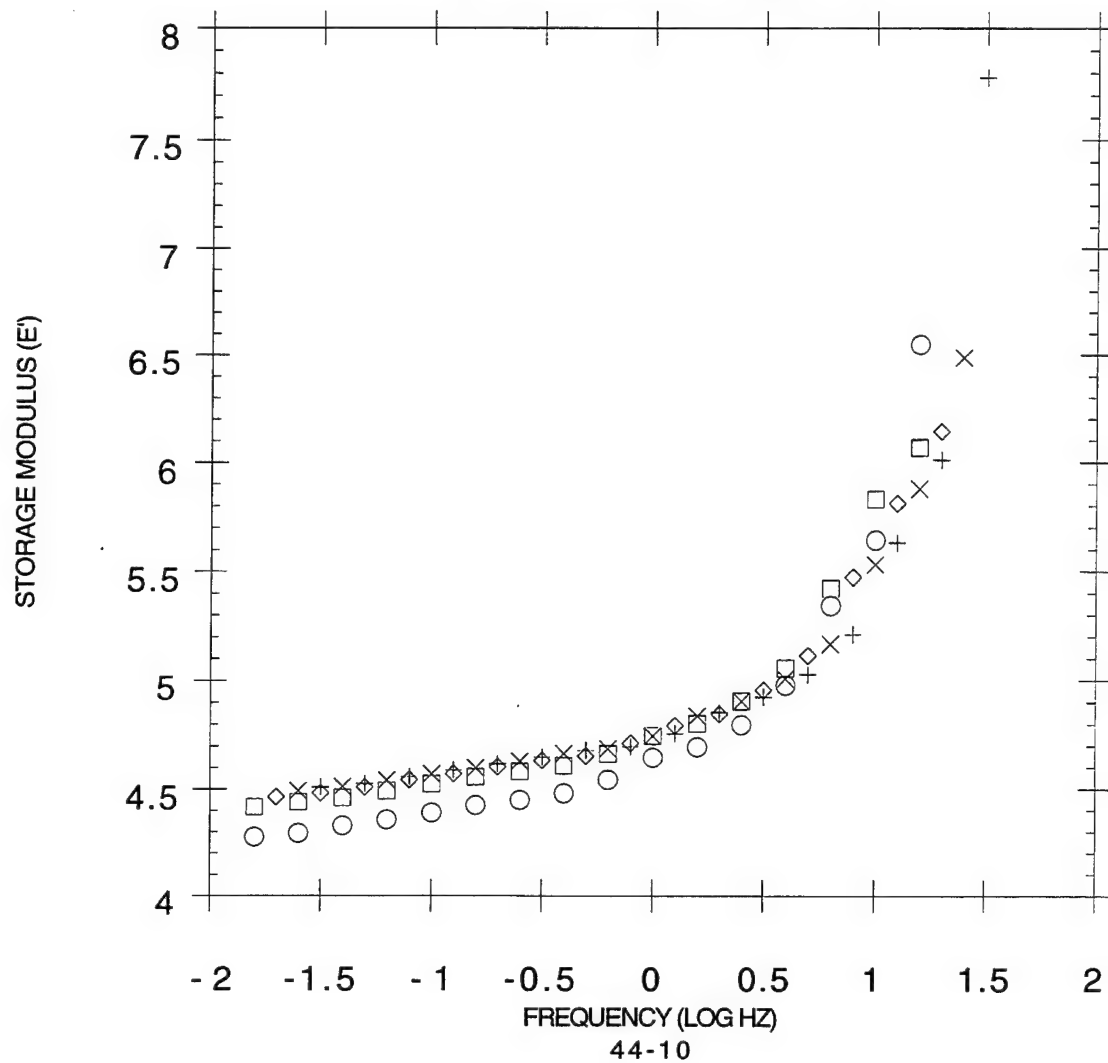




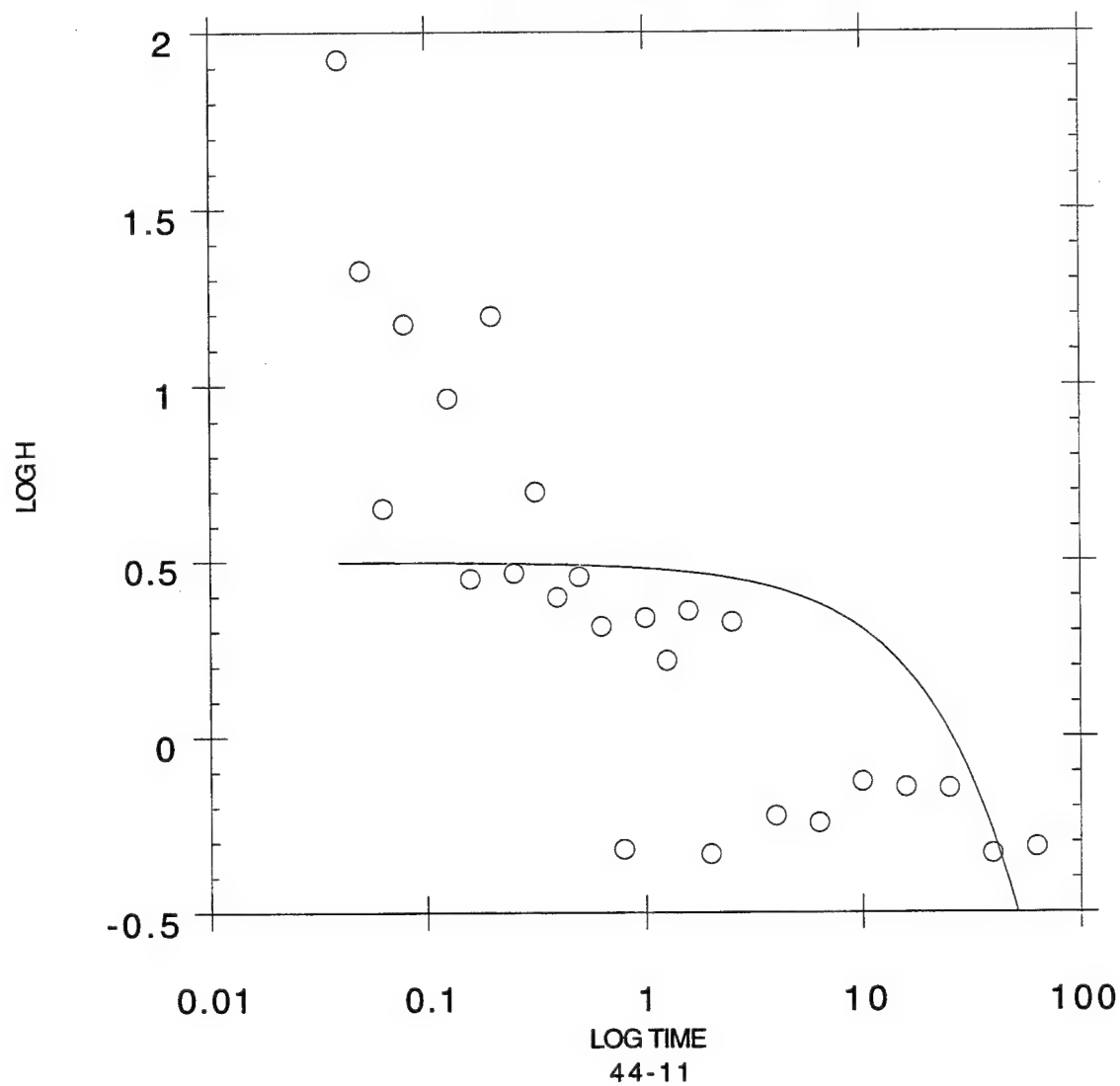
As-Cured 300 C



Post-Cured 250 C



Post-Cured 250 C



Discussion of Results :

Overall, the graphs for the behavior of AFR700B's storage modulus versus frequency turned out as expected. After a prolonged period of time it can be observed that the storage modulus increases as a function of time in a parabolic manner. Previous experiments by Ferry³ indicate that eventually this pattern experiences a slope inflection and levels out at a maximum storage modulus limit.

The only unexpected peculiarity that these results display occurs between the starting point of the test and the end of the first hour. During this time interval, the storage modulus seems to jump unexpectedly to higher values while preserving the same slope ratio as other hours. The reason for this jump may lie in a chemical transformation that occurs between the beginning of the experiment and the termination of the first hour.

The behavior of relaxation spectra, on the other hand, provided less useful results. Almost all of the data points (as can be seen from the H' graphs) appear scattered and disorganized. Several plausible reasons for these chaotic results include 1) the load force was too small, thereby amplifying experimental noise, 2) the Dyna-Stat machine is not precise enough to measure storage modulus data, 3) the equation use in determining relaxation spectra was inadequate or not fully utilized (such as assigning the variable "A" to equal 1).

Conclusions :

- 1) In general, storage modulus increases with an increase in frequency for AFR700B advanced composite material
- 2) The relaxation spectrum for AFR700B advanced composite material tends to decrease as a function of superpositioned time
- 3) Both storage modulus verse frequency and relaxation spectrum graphs appear to be nearly identical for As-Cured 250 C runs, As-Cured 300 C runs, and Post-Cured 250 C runs
- 4) This current method of developing relaxation spectra for AFR700B advanced composite material was not precise enough to obtain smooth, accurate results during these experiments

Acknowledgments _:

I would like to express my sincere thanks to Katie Thorp, Dr. David Curliss, Brian Rice, Ken Lindsay, Bill Ragland, Jim Lute, Dr. Dave Anderson, Capt. Dave Ciminelli, John Russell, Greg Nichols, Joe Scheckel, and Ryan Q. Simon for their patience and assistance with this research project.

References _:

1. Mel Schwartz, Composite Materials Handbook 2nd edition, McGraw-Hill Inc., US (1992).
2. Katie Thorp, Hygrothermal Stability of Polyimide Resins and Composites, University of Dayton Research Institute, (1995).
3. John D. Ferry, Viscoelastic Properties of Polymers, 3rd edition, John Wiley & Sons, Inc., US (1980).
4. L.C.E. Struik, Effect of Thermal History on Secondary Relaxation Processes in Amorphous Polymers, Polymer, Volume 27, Netherlands (1986).
5. L.C.E. Struik, The Mechanical and Physical Ageing of Semicrystalline Polymers: 1, Polymer, Volume 28, (1987).
6. Ryan Q. Simon, The Combustion of Advanced Composites, High School Apprenticeship Program, Armstrong Toxicology Laboratory, Area B WPAFB, August (1994).

Capacitor Characterizations

Rebecca A. Smith

Kettering-Fairmont High School
3301 Shroyer
Kettering, OH 45429

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

CAPACITOR CHARACTERIZATIONS

Rebecca A. Smith
Kettering-Fairmont High School

Abstract

The characteristics of capacitors undergoing different conditions were observed. Many different kinds of capacitors, including diamond-like carbon, film, ceramic and electrolytic were studied. The capacitors were subjected to a frequency scan in which the frequency was incremented from 20Hz to 1 Mhz and capacitor, dissipation factor, equivalent series resistance and other readings were taken at each interval. In addition, specific capacitors were placed in a high temperature environment for 1 week and the insulation resistance readings were taken.

Introduction

Capacitors are found in almost every electronic device from the simplest circuits to complex computer systems. Unfortunately, they are also the most vulnerable component and can be damaged when the circuit undergoes conditions such as frequency changes, high voltage conditions or high temperatures. We tested many different types of capacitors with characterization equipment to discover the trends under different conditions, including the ones mentioned above. We measured capacitance, dissipation factor and equivalent series resistance (ESR) on the majority of the capacitors tested. We are presently finishing up a long range experiment to test the effects of heat on a new kind of multilayer ceramic capacitor (MLC). There are 13 capacitors in the set; five AVX48-A capacitors, 5 AVX48-C capacitors and 3 AVX capacitors (AVX is the company that manufactures the capacitors). For these caps, we ran frequency scans and tested the dissipation factor, capacitance and ESR. We then placed the capacitors in the oven for a week. After taking them out, insulation resistance (IR) readings were taken and the process began for the next week. In addition, capacitor testing was performed on capacitors made by Northrop, the University of Cincinnati and other companies. Each set of capacitors were subjected to a unique set of tests. The Northrop capacitor testing consisted of a series of 7 tests, including 2 thermal scans, a frequency scan at high temperatures, and changes with a bias voltage applied to the capacitor. The DLC capacitors underwent a breakdown test to see at which voltage the capacitor no longer performs correctly.

Methodology

For most of my term, I used the capacitor characterization system. I learned about the different tests run on capacitors to test their durability and performance. Presently we are testing a series of ceramic caps that may be implemented into future aircraft systems that depend on electronic instead of hydraulic based controls.

We ran tests using an LCR meter, and a computer using the CAPDAS (Capacitor Data Acquisition System) program to acquire our data. In addition, we used a series of different test leads on the capacitors. For example, we tested different kinds of DLC (diamond like carbon) capacitors for the University of Cincinnati using a test stand with two probes touching the different contacts on the capacitor. For most capacitors, however, we used two clips, one on each lead of the capacitor.

The computer required an open measurement and a short measurement in order to accommodate the minor differences due to outside factors such as cable length and quality of the test leads. Unless the leads were changed, the tests could be performed continuously using those corrections.

The capacitor was hooked up to the test leads, and the capacitor data acquisition system (CAPDAS) was started. This program performed the frequency and voltage scans and controlled the oven on the thermal scans; as a result the data was very accurate. The computer took 10 readings for every increment, and used the average of these readings to plot a graph at the end of the test.

In special cases, specifically the long range tests being made on the MLCs, the capacitors were placed into a thermal aging oven at 200 degrees Celsius for a week. They were taken out and IR tests performed. Insulation resistance was measured on a separate apparatus and in a less accurate method than the frequency scans. The capacitors were hooked up to a 250 volt power source, and a timer started when the power was turned on. The capacitors charged for 5 minutes, and then the resistance was measured using an a high resistance meter. Due to outside noise, such as fluorescent lights, the needle on the meter never settled down completely, and a best guess reading had to be made. The capacitor was allowed to discharge for a few minutes, was shorted out, and the test run again. Finally, the average of the 3 trials was

recorded as the final IR reading. In order to view the trends of IR over 20 weeks, the numbers were inserted into a spreadsheet and charted. Microsoft's Excel was the ideal program to use for these charts and as a result, I am extremely comfortable with the software and was able to assist others in my lab with the software.

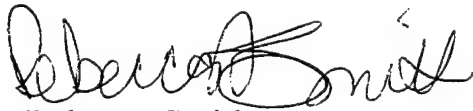
The MLCs were taken to a different facility, and tested using the x-ray diffraction machine and SEM. I was unable to use either machine directly because I wasn't properly trained on either, but I was able to see how a sample is mounted and what preparations need to be made to perform both types of tests.

Two different sets of x-rays were taken with the x-ray machine; one set with the capacitor mounted in the x-ray machine, and one set with the ceramic powder (mostly composed of barium titanate) mounted inside. The SEM was used to take pictures of the layers of the capacitor. I was unaware that there were up to 30 layers of ceramic dielectric included in one capacitor; a fact I was able to witness by viewing the pictures taken by the SEM.

Summary

I was unaware that the characteristics of capacitors depended so much on other factors like frequency, voltage and temperature, and that capacitors underwent such rigorous testing before they were implemented into applications. Coming into the program, I had only read about and used equations for capacitors, but by being in contact with capacitors continuously I understand them and their applications a lot better than I ever could by just learning equations. Having the tests explained to me as I was performing them helped me to understand how the capacitors worked and why they behave the way they do. In addition, I was exposed to advanced equipment, such as the SEM, x-ray machine and sputtering machine, that I may not see again until I'm a graduate or Ph.D. student. I will probably never study capacitors as extensively as I have this summer, but I know I will see this type of testing and characterization equipment in my future as an engineer.

I have no inventions to report as a result of my work on the AFOSR 1995 Summer Research Program.

A handwritten signature in cursive script, appearing to read "Rebecca Smith".

Rebecca Smith

95-0693

8-9-95

**EXPERIENCE AT THE AIRCRAFT SURVIVABILITY AND SAFETY
BRANCH AT WRIGHT-PATTERSON AIR FORCE BASE**

Nick Steffano
High School Apprentice
Aircraft Survivability and Safety Branch

Kettering Fairmont High School
3301 Shroyer Road
Kettering, OH

Final Report For:
High School Apprenticeship Program
Wright Laboratory

Sponsored By:
Air Force Office of Scientific Research
Bolling Air Force Base, DC
and
Wright Laboratory

August 1995

EXPERIENCE AT THE AIRCRAFT SURVIVABILITY AND SAFETY BRANCH AT WRIGHT-PATTERSON AIR FORCE BASE

Nick Steffano

High School Apprentice

Aircraft Survivability and Safety Branch

Wright-Patterson Air Force Base

Abstract

In the Survivability and Safety Branch of Wright Labs, aircraft fire suppression and the study of composite materials are the two major efforts in progress. The aircraft fire suppression group is trying to find an adequate replacement for halon. This is due to its harmful effects to the atmosphere. Composites are also being researched at this facility because their usage is becoming more prevalent in today's aircraft. Because composites are unlike metals, they behave differently when impacted by conventional weapons. The results from test shots, which are done at one of four gun ranges, on the composite panels are taken and put into a spreadsheet for further analysis. The apprentice learns just as much about the real working world and about the career they are considering as he does about the project they are working on.

EXPERIENCE AT THE AIRCRAFT SURVIVABILITY AND SAFETY BRANCH AT WRIGHT-PATTERSON AIR FORCE BASE

Nick Steffano

Two of the major efforts at the aircraft survivability and safety branch in the flight dynamics directorate include: 1) the testing of composite materials and 2) testing numerous types of fire suppression agents that will eventually replace halons.

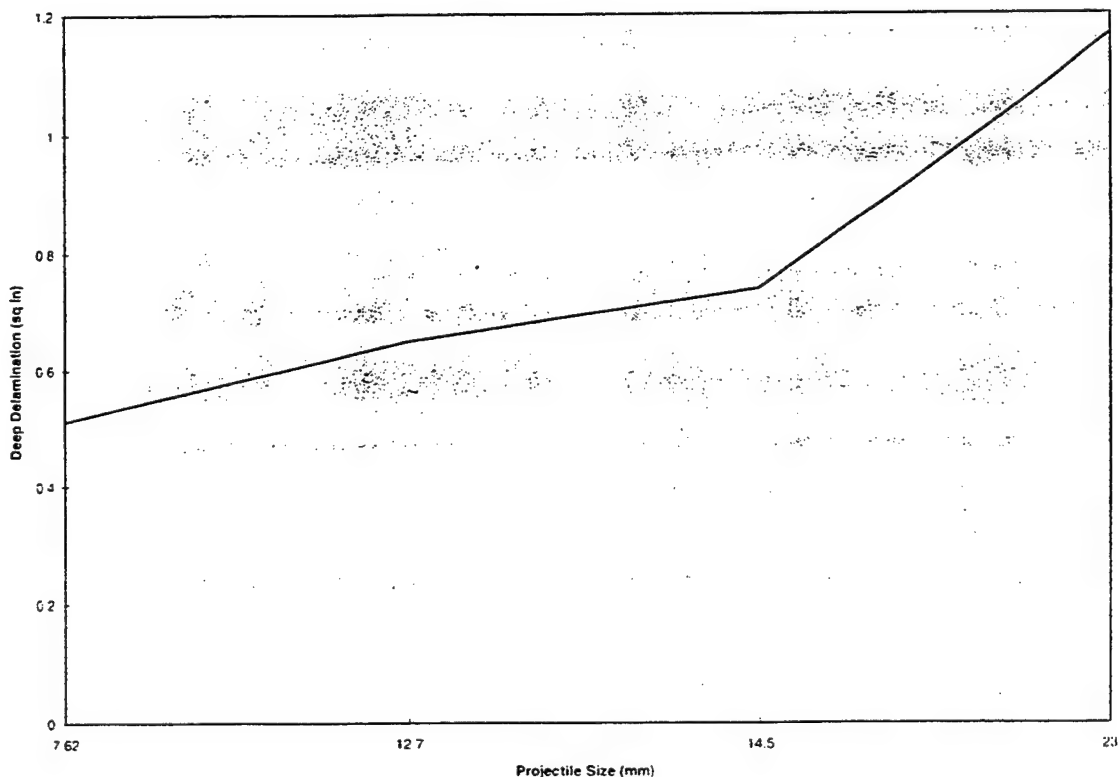
A composite material is made of layers of materials such as graphite, kevlar, fiberglass, or other types of material. Each layer is composed of strands of the material that are thinner than a piece of human hair. Between each layer is a resin that is used to bond them together. The layers are stacked on top of each other so that the angles of the fibers on the surrounding layers are different. For example, one layer may be laid so that the fibers are at a zero degree angle. The next would be laid at a forty-five degree, then the next at a ninety, and the last at a negative forty-five. After the four layers have been laid, the material layer order can be mirrored or it can start the sequence over again. The number of sequences used is dependent upon the thickness of the composite panel you want to achieve. Once the final panel is put together, it is placed into an autoclave. The autoclave is heated to the desired temperature and set to the desired pressure to set the resin for good.

Testing the composite panels, as with testing anything else, involves many different steps. First a test plan must be written. The test plan explains the purpose of the research, where it will be done, and the details of how the test will be run. Once completed the test plan is handed over to the people in charge of actually putting the equipment for the test together. This includes the contractor that will set all of the sensors, and will prepare a fixture for the target. Some of the instruments that are used are break paper, light screens, wire triggers, electromagnetic sensors, heat sensors, and pressure sensors.

When all of the preparation is done, the testing can begin. The testing involves shooting each panel with an assortment of enemy projectiles to simulate a real battle, which include a 7.62mm, 12.7mm, 14.5mm, and 23mm round, or fragments. Each different round can have either a High Explosive Incendiary (HEI) or an Armor Piercing Incendiary (API) tip placed on it. Other variables included in the testing are projectile velocity, angles of impact (obliquity and yaw), and number of layers on the target panel.

After completion of the tests, the data is taken and entered into a spreadsheet on Microsoft Excel. Examples of the data taken include: 1) damage that is done to the panel 2) whether or not there is a flash or fire from the impact. Then the data that have the same variables are separated and grouped together. For example one of the groups may contain the following variables: a 7.62mm API, 12.7mm API, 14.5mm API, and a 23mm API projectile all impacted a 48 layer panel at about 1500 fps at a 30 degree angle of obliquity. One thing that can be done with this data is to create a chart. Charts can be made that plot data such as the size of the projectile versus how much deep delamination it caused, as shown below. With these charts, one can see the relationship between the size of the projectile and the damage that the panel incurs.

Projectile Size vs. Deep Delamination



Deep delamination is one type of damage that can be done to composite materials. It is when some layers of the panel separate from each other. Many times this type of damage is invisible. Delamination could occur by simply dropping a hammer onto a wing. One way to determine it is to do a C-scan on the panel. A C-scan is a process of putting the damaged panel into a tank of water, then sending sound waves through it. The waves then bounce back and are collected in a computer. The computer then calculates the time that it takes to retrieve the waves and creates a digital picture of where there has been delamination. The picture is printed out on sheets of paper that can be up to three feet by wide by four feet tall, but are usually scaled down to be more easily manipulated. To find out the area of delamination a planimeter is used. This is an instrument that measures the area of a plane figure as a mechanically coupled pointer traverses the figure's perimeter.

Once all of the data has been entered into a database and analyzed it can really be put to use. Using the data, aircraft design can be modified, techniques for battle damage repair can be made, and a general idea of what will happen when a composite is struck by a projectile can be made. This knowledge can be put into use in the battlefield because a repair technician will know what type of damage is most frequent, and therefore will not have to carry repair material that is needed very rarely. It can instead only carry what is mostly needed. The collected data can also be used to derive vulnerability equations. This is most helpful in deciding what plane would be best to fly a particular mission. Some planes may be more vulnerable on their underside than others. Therefore those aircraft would not be good for a low altitude mission.

Another major effort of study in this branch is to find a fire suppression agent to replace halons. Halon (CF_3Br) is not environmentally safe. It is slowly depleting the ozone layer. There are two different types of testing being conducted for fire suppression. The testing at the gun range is similar to that of composites because they must shoot through a dry bay and create a fire. The other testing is at a test facility where fuel is sprayed into an engine nacelle and then ignited by something similar to a spark plug. The fire is then put out, and the residual fuel is watched closely to see if it will re-ignite itself

from a hot plate that has been set up inside the engine nacelle. Once a fire has been established, a model of a fire suppression system that may be in an aircraft attempts to extinguish the flames. Obviously the goal is to put out the fire, but there are other things that must be kept in mind while achieving this. Examples are: 1) where the actual suppression system will have to be implemented 2) what is the least amount of agent that it takes to put out the fire 3) what agent will be dispensed 4) how it will be dispensed. The fire suppression group is also researching better methods of detecting fires so that the agent will be released at the proper time.

As an apprentice here at the Aircraft Survivability and Safety Branch at Wright Labs for eight weeks I have gathered an incredible amount of knowledge, not only about aircraft survivability and safety, but also about the real working world. I have become very efficient on the computer using such programs as Microsoft Excel and Microsoft Word for Windows. I have even had the opportunity to learn how to use a planimeter, which is so out of date that it isn't even made anymore. About two weeks from the end of my stay here, I was given the task of writing and conducting my own test on ablatives or fire retardants. I was successful in completing the test plan and collecting some of the pretest data, but unfortunately my time here expired before we were able to complete the shots on the panels. Everyone here has been willing to talk to me about my future in college and about my career as an engineer. There is an employee that has graduated from almost every college that I am considering applying to working in my building. Although at times it was a struggle getting up during the summer every day at 7:00 a.m., I am truly glad I was involved in this program.

I would like to thank Ms. Patricia Pettit and Mr. Alex Kurtz for everything, and I hope to see them again next year.

INSTALLATION AND APPLICATION
OF AN IEEE 488 COMPATIBLE PC INTERFACE
AND
THE INSTALLATION, APPLICATION, AND MEASUREMENT
OF MICRO-MACHINED ACCELEROMETERS

Todd Stockert

Centerville High School
500 W Franklin Street
Centerville, OH 45459

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

INSTALLATION AND APPLICATION
OF AN IEEE 488 COMPATIBLE PC INTERFACE
AND
THE INSTALLATION, APPLICATION, AND MEASUREMENT
OF MICRO-MACHINED ACCELEROMETERS

Todd Stockert
Centerville High School

Abstract

This particular lab assignment entailed the interfacing of two digital accelerometers. This consisted of three phases. Phase one included programming of a PC to handle and distribute commands and data over the IEEE 488 bus. Phase two included building of two accelerometer circuits and interfacing them with the IEEE 488 bus for evaluation of sensitivity. The third phase incurs the use of this technology in various applications.

Introduction

The lab environment is very ready for automation. Many devices often must be handled together--a task that can become cumbersome. Many pieces of lab equipment have incorporated a standard IEEE 488 bus interface into their systems. This allows a port of communication between devices and computer allowing easy interface and simplification of procedures. The problem is the actual control of the bus by PC. The commands are intricate and repetitive, often making complex tasks cumbersome with the normal command prompt. A user friendly, adaptable, expandable, graphical interface is necessary. The Windows based IEEE 488 interface program (Win488) developed during this project fulfills these requirements. The interface program was tested on the analysis of the XL50 accelerometer. The XL50 is from the most recent developments in Micro-Machining technology. The XL50 is an accelerometer surface rendered from one piece of silicon. The possibilities for application of such a device are numerous, but first the device must be tested for sensitivity and consistency. Tests were run on the XL50 and results confirm the static reliability of the device.

Equipment

The equipment used included two accelerometers, a signal analyzer, and a 386 PC. The accelerometers used during the testing were two Analog Devices XL50 TO-100 mounted micro-machined sensors. Unlike traditional accelerometers, the ADXL50 is surface-rendered from one piece of silicon. Traditionally a proof mass was used to carry out acceleration measurements, this new technology uses suspended silicon in a variable capacitance system to observe changes in forces(Figure 1). The advantages of these devices over previous models are numerous. Weight considerations and mounting problems are eliminated as these sensors can be mounted directly on a circuit board as easily as a transistor. Secondly the cost of fabricating, and consequentially obtaining the devices is extremely low at around five dollars per sensor. The signal analyzer used was the B&K model 2032. The B&K 2032 contains an analog to digital convertor which was used to interpret the voltage data from the ADXL50s. The data could then be analyzed on the B&K 2032 or sent over the incorporated IEEE 488 bus to a PC.

Procedures

The initial phase of the project included the general interfacing of the IEEE 488 bus to the computer. This was accomplished using Turbo Pascal for Windows. The Object Oriented program style was studied and implemented using resource libraries and numerous guidebooks. A general Windows interface was achieved with the standard compilation of user-friendly pushbuttons and dialog boxes(figure 2). A help interface was compiled along with the program to facilitate an even friendlier

environment. The IEEE 488 bus was controlled using standard conventional file-control techniques through 'ASSIGN', 'READLN', 'WRITELN' and other such procedures. An excerpt of the source code from a dialog box control object is included in Table 1. The ease of expansion is readily apparent, as relatively few commands are necessary to implement a new command. It is organized into various units with one control unit used to modify IEEE 488 actions and a main program to initiate these procedures. If a procedure PROC A were needed to give a 'Proceed with averaging' command to the B&K 2032, the modification would be what is listed in Table 2. The control element would be a pushbutton in the main window. The program was expanded in this way to create all the necessary elements for testing the XL50 over the IEEE 488 bus. Three other methods of expansion are present in the Win488 program. The TPC.EXE Turbo Pascal compiler is accessible through the Win488 program to recursively compile user created text-based procedures. A template (TEMPLATE.IOF) file is included for reference. The user also has the option to run any executable file from within Win488. This allows configuration without a Turbo Pascal specific compiler. The standard command prompt interface program (KYBDCTRL) is also included in Win488 for less complex infrequent tasks.

The next phase of this project required the configuring of the AD XL50 Accelerometer for testing. The circuit necessary to evaluate the XL50's performance is not complex. Most of the necessary wiring is on board the accelerometer. The major outboard components are the zero-g set system, the power demodulator, and various optional filter systems. The circuit used was the standard schematic recommended by Analog Devices and the components were selected for an approximate ten g range(Figure 3;Table 3). The range selected allowed for a 200 millivolt change per g. The full range of the device is fifty g in either direction along its axis and it can withstand shocks up to 2000 g. To make the system more usable, a second sensor was added to allow the processing of differential measurements. The two sensors were hard-wired on separate platforms to allow differential mobility. The sensors were then hooked up to a National Semiconductor LM3900 in an inverting amplifier configuration for difference determination(Figure 4). A switch was later added to allow the user to choose whether both sensors read out or whether the inverter took one of the inputs.

The final phase included the interpretation of data obtained by the accelerometers. The accelerometers created an analog voltage from pin 9 which had a zero g level at 2.5 V and increased or decreased proportionately with reference to acceleration. In the first phase of testing this voltage was sent to the B&K 2032, converted to a digital signal, manipulated and eventually sent over the IEEE 488 bus to the PC. The measurement setups used were the Cross-spectrum analysis, Time-

dependent spectrum averaging, and the Cepstrum. This study centered around the first two, but the Cepstrum could be easily included for more in depth causes, such as finding the differential 'where', 'when', and 'what'. The Cross-spectrum analysis allowed experiments to be performed over large amounts of time with a relatively small amount of random error. The bandwidth used in this study was between 15.6 mHz and 1.56 Hz. This was chosen to skip over the bulk of most non-linear activities. Averaging was set to 32767 iterations, the limit of the B&K 2032(Figure 5). The Time-dependent spectrum averaging was useful in determining the actual noticeable sensitivity of the XL50. The results of the experiments were then analyzed through the application of various on board functions in the B&K 2032. The meaningful plots could then be downloaded via the Win488 program and graphed through GNU PLOT.

Results

The results from the various phases were mixed, though optimistic. During the programming phase, most of the goals originally desired were met. A user friendly expandable program was created. It was given the ability to send commands to the B&K 2032 or any other IEEE device through a simple visual interface. The program was able to download the relevant data from the analyzer. The transfer method is unfortunately slow. Fast DMA block transfer created numerous errors, so Win488 reads each data point separately. In iterations under 500 this is manageable, but as the number of points rise, so does the time for transfer. Plots with 800 points took over 3 minutes. Other methods of transfer should be looked at and tested. Another minor problem concurrent with the programming occurred in the interfacing of window control elements. There was difficulty in creating the code necessary to control secondary controls, such as controls within dialog boxes not initially displayed in the main window. This was cleared up through an auxiliary type statement and additional control procedures, allowing complete operation of secondary controls. This allows for the manageable addition of control objects for future tasks and devices. The program should be well suited for the lab experimentation environment. The XL50 accelerometers also had a mixture of success and frustration. The devices themselves show enormous possibilities. After staying active for more than 96 hours straight, they showed no zero-g level changes. The circuit responded consistently to stimuli. Repetitive averages and comparisons displayed a few omni-directional vibrations picked up as accelerations, along with a few vector type accelerations. The causes of these disturbances could not be positively identified, but building vibration, large equipment systems, and earth rotation were all ruled as possibilities. Under a bandwidth of 800 Hz, motions among the order of people walking down a hallway were discerned through use of the analyzer's on board integration functions. Exact

sensitivity was not determined due to imprecise mounting, extraneous vibrations, and inexact measuring techniques. A possible problem in measurements with the XL50 is the drift error published at .1g. The use of two devices may have cancelled the error, but the precision of such measurements should be reexamined. Further testing is planned. A rate table has been acquired and an experiment is currently being developed to test the quantitative viability of the sensors. The broad testing that was performed did reasonably confirm the XL50 was measuring accurately.

Applications

The Win488 interface program developed could be easily expanded for future use in many applications outside of the XL50 realm. A tentative use could be to help interface a test oven for evaluating heat dependent components such as transistors. A general application that could be gradually obtained is a complete lab control system. This would entail control and data access to a large database of devices. If this were successful, remote experimentation could be performed over network or modem and duplicate equipment costs could be avoided. The XL50 sensors are also very general in their abilities. The current models could be built upon for use in a completely electric navigation system free from the bulk of current systems. The heavy gyroscopic centered systems could be reduced from many pounds to a few ounces, from a large black box to a system no larger than a pack of cards. The accelerometers could also find themselves into commercial markets. They are already in airbag systems for many automobiles. They could be placed in manufacturing machinery to detect subtle problems. They could be placed in shoes to check strain levels. As the technology grows, sensors could be placed in humans to monitor reoccurring health problems, telling a patient when something is wrong. Many applications lie ahead for this new and relatively inexpensive technology.

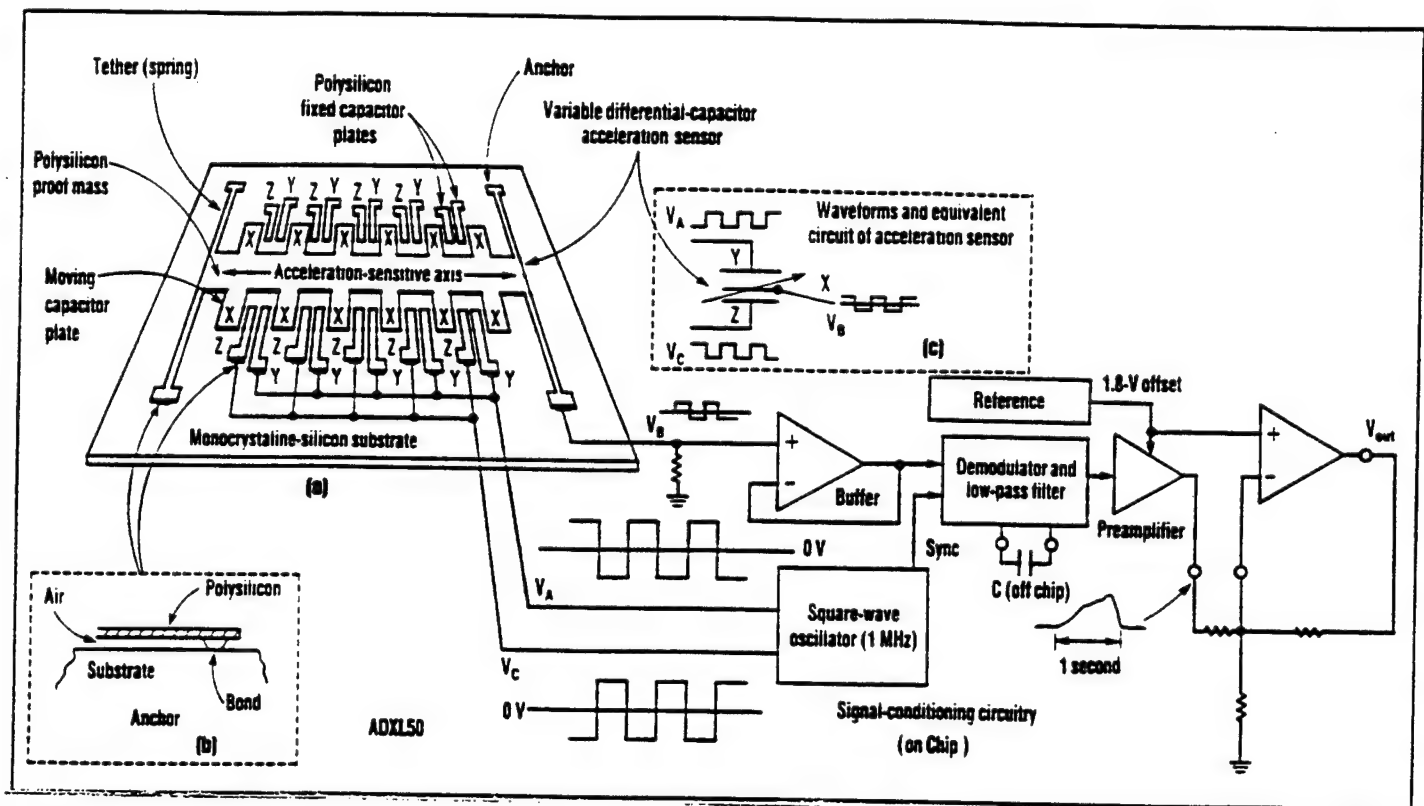
Conclusions

The purpose of work done on this project was threefold; to create an interface to the IEEE 488 bus, to connect the XL50 into an observable test circuit, and to test the functioning of the XL50 accelerometer through the B&K analyzer using the interface program. Each task was completed successfully and will hopefully be expanded upon in the future. The program developed can be easily expanded in the future to include many devices outside of the B&K 2032. The circuitry used for the XL50 was not new, but was confirmed as stable and feasible for future use.

References

- Brüel & Kjær. 2032 Dual Channel Signal Analyzer Type 2032 Instruction Manual.
September 1983.
- Borland International. Turbo Pascal for Windows. Windows Reference Guide. 1991.
- Borland International. Turbo Pascal for Windows. Windows Programming Guide. 1991.
- IOtech. Personal488 User's Manuel. Revision 2.1. December 1988.
- Analog Devices. Monolithic Accelerometer with Signal Conditioning [device description].
Revision 0.

Figure 1: Layout of the Analog Devices XL50 Accelerometer



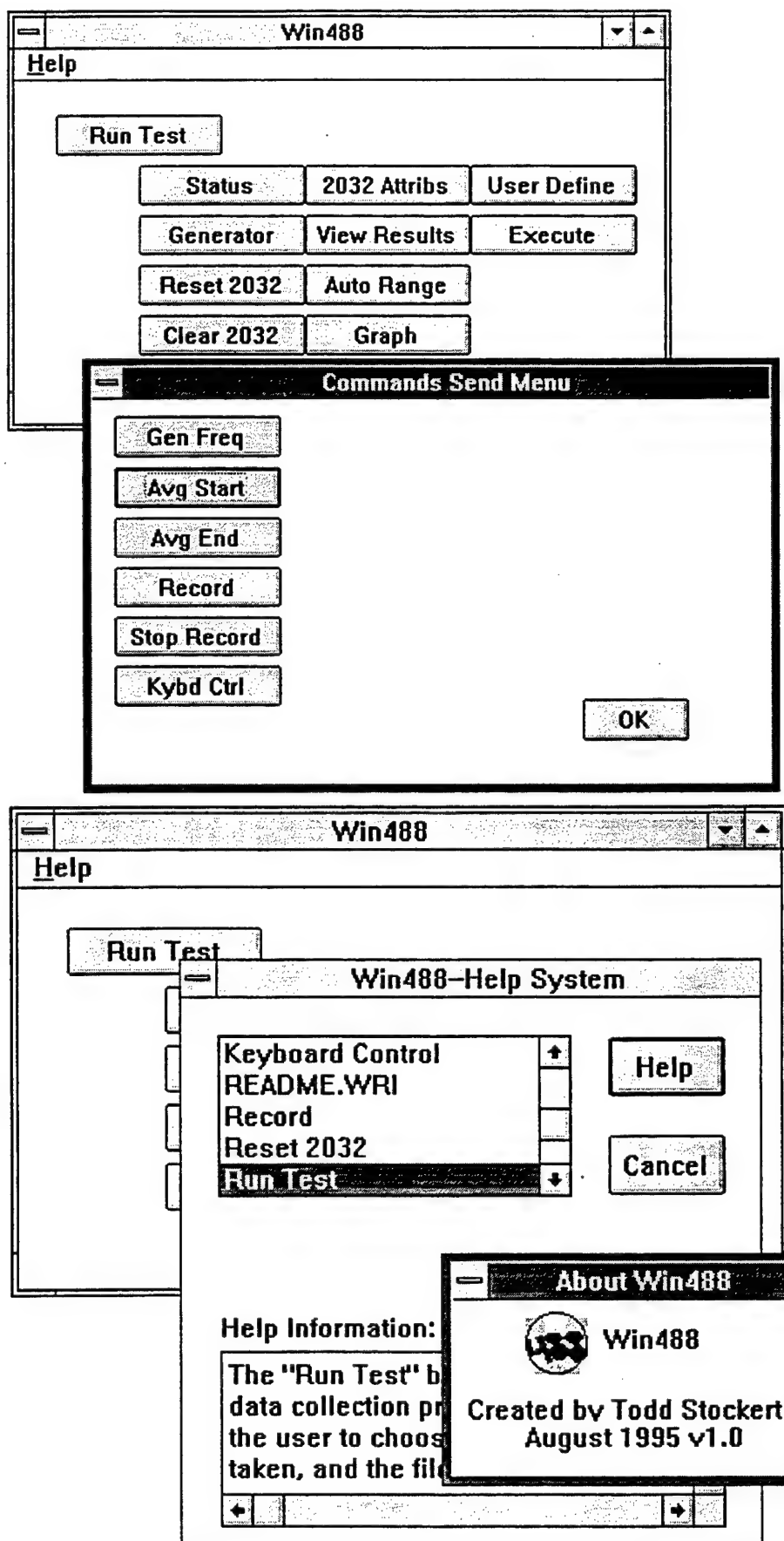


Figure 2: Window Environment of Win488
47-9

Table 1: Source Code from Dialog Control Objects within Win488

```

(**in program Win488**)
Type      {(2032 Attrib) Dialog}
pmydialog1:=^tmydialog1;  {Box Object Type Statement}
tmydialog1=object(tdialog)
  procedure genfreq1;
  procedure genfreq(var msg:tmessage);
    virtual id_first + id_genfreq;
  .....

  procedure averagestop(var msg:tmessage);
    virtual id_first + id_avgend;
  procedure kybdctrl(var msg:tmessage);
    virtual id_first + id_kybdctrl;
    {****add function here****}
end;
.....

{****Dialog box Procedures****}
procedure tmywindow.defineattr(var msg:tmessage);
var
  returnvalue:integer;
  d:pdialog;
  p1:pbutton;  {Defines active parts of the}
begin  {COM_SEND resource for use in the Win488 program}
  d:=new(pmydialog1, init(@self, 'COM_SEND'));
  new(p1, initresource(d, id_genfreq));
  new(p1, initresource(d, id_avgstr));
  new(p1, initresource(d, id_avgend));
  ...
  returnvalue:=application^.execdialog(d);
end;
...

procedure tmydialog1.averagestop(var msg:tmessage);
begin
  openiofiles('logfile.dat');
  sendavgend;
endproc;
end;

procedure tmydialog1.kybdctrl(var msg:tmessage);
begin
  winexec('c:\ieee488\kybdctrl.exe',sw_normal);
end;
.....

(**in unit ct2032nv**)
Procedure sendavgend;
Begin
  writeln(ieeeout,'OUTPUT 26;cp ck');
end;

```

Table 2: Modifications Necessary for Win488 to Create a New Procedure

Add to type statement:

```
Procedure PROCA(var msg:tmessage);
  virtual id_first + id_proca;
```

```
  Add to tmywindow.init:
pb1:=new(pbutton, init(@self, id_proca, 'Avg Proceed',
  350,110,100,24,false));
  ^X ^Y ^W ^H
```

```
  Add to procedures section of Win488:
Procedure tmywindow.proca(var msg:tmessage);
begin
  openiofiles('logfile.dat');{no data is sent to logfile.dat}
  sendproca; {the filename is necessary to fulfil a variable}
  endproc; {If the procedure does read data, a valid}
  {name must be entered, or a call to procedure namelog}
  {must be asserted}
```

```
  Add to unit ct2032nv.pas:
{in procedure list section}
Procedure sendproca;

{in implementation section}
Procedure sendproca;
Begin
  writeln(ieeeout,'OUTPUT 26;cp bk');
end; {sends 'cp bk' to device 26}
  {which in this case causes the}
  {B&K2032 to proceed with}
  {averaging}
```

Table 3: Recommended Resistor Values for Setting the Circuit of Figure 3 to Several Common Full-scale Ranges

FS (g)	Buffer Gain	SF in mV/g	R1	R2	R3
±50.0	2.11	40	49.9 k	105 k	100 k
±40.0	2.63	50	39.2 k	103 k	100 k
±30.8	3.42	65	40.2 k	137 k	100 k
±26.7	3.95	75	28.7 k	113 k	100 k
±20.0	5.26	100	26.1 k	137 k	100 k
±10.0	10.53	200	23.7 k	249 k	100 k

The schematic diagram illustrates the internal circuitry of the ADXL50 accelerometer. It features two operational amplifiers: a PRE-AMP and a BUFFER AMP. The circuit is powered by a +5V supply, which is decoupled with a 0.1μF capacitor. The input stage includes a 0.022μF capacitor (C2) connected to pin 4 and another 0.022μF capacitor (C1) connected between pins 2 and 3. Pin 5 is connected to ground (COM). The reference voltage (VREF) is applied to pin 6 via a 50kΩ trimmer, labeled '0g LEVEL TRIM', which is also connected to ground (V_x). The pre-amplifier's output (pin 8, V_{PR}) is biased at +3.4V and connected to pin 10 (V_{IN-}) through resistor R1. The non-inverting input of the buffer amplifier (pin 9) is biased at +1.8V and connected to pin 10 through resistor R2. Resistor R3 connects the output of the buffer amplifier (pin 9, V_{OUT}) back to pin 10. The output of the buffer amplifier is also connected to pin 1, which is biased at +5V and decoupled with a 0.1μF capacitor. The circuit is labeled 'ADXL50'.

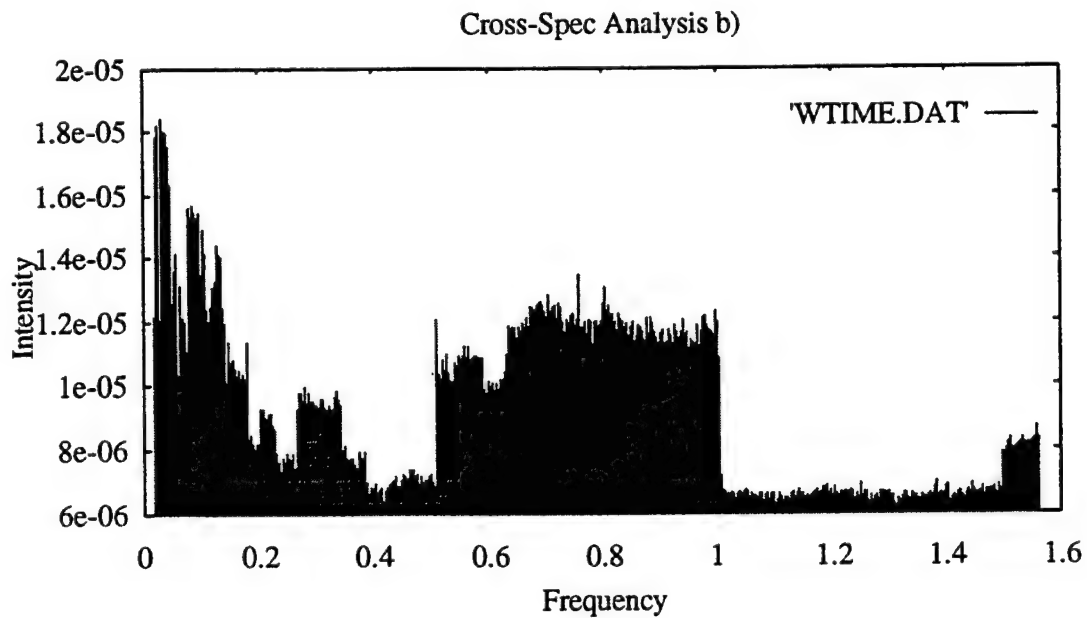
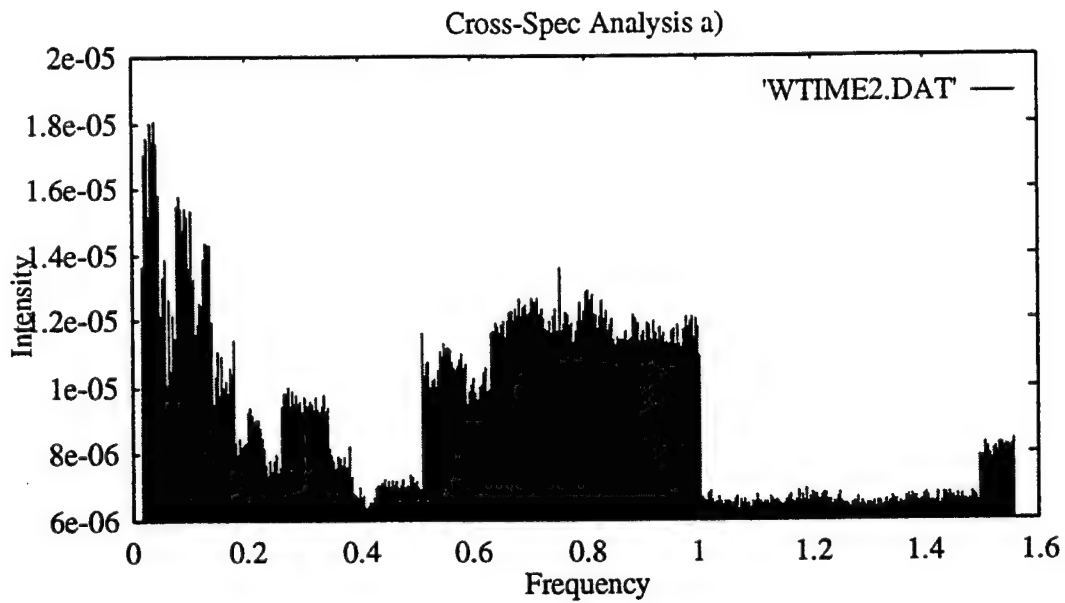


Figure 5: These Cross-Correlation Plots reveal the presence of omni-directional and directional disturbances at specific frequencies(hz). a) measurements were taken 90° from the b) measurements.

**An Investigation of InAs/In_xGa_{1-x}Sb Band Gap
Properties with Random Layer widths**

Jeroen W. Thompson

**Beavercreek High School
2660 Dayton-Xenia Rd.
Beavercreek, OH 45430**

**Final Report for:
High School Apprentice Program
Wright Laboratory**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC**

and

Wright Laboratory

August 1995

AN INVESTIGATION OF $\text{InAs}/\text{In}_x\text{Ga}_{1-x}\text{Sb}$ BAND GAP
PROPERTIES WITH RANDOM LAYER WIDTHS

Jeroen W. Thompson
Beavercreek High School

Abstract

Using the transfer matrix technique, a model of the $\text{InAs}/\text{In}_x\text{Ga}_{1-x}\text{Sb}$ superlattice was developed. This was used to study the effects of random layer width variations on the band gap and the cut-off wavelength through a Monte Carlo approach. The results indicate that a greater than 1 monolayer control over the growth of $\text{InAs}/\text{In}_x\text{Ga}_{1-x}\text{Sb}$ is required to maintain uniform detector response.

AN INVESTIGATION OF InAs/In_xGa_{1-x}Sb BAND GAP PROPERTIES WITH RANDOM LAYER WIDTHS

Jeroen W. Thompson

I. Introduction

Advances in growth processes, especially molecular beam epitaxy, have allowed the production of many esoteric materials, such as InAs/In_xGa_{1-x}Sb, whose electrical properties can be tailored to meet various needs. One such use of InAs/In_xGa_{1-x}Sb lies in infrared detectors. The parameters given by Heller, *et al.*, for the optimization of absorption for InAs/In_xGa_{1-x}Sb may allow the use of this substance in optical arrays¹. However, maximum absorption in the far IR range requires the use of extremely thin layers on the order of a few dozen atoms or less. Any variation in the width of these layers could have great adverse effect on both the amount of absorption, and which wavelength of light actually is absorbed. This problem necessitates the study of the effects of variations in layer widths.

II. Discussion of InAs/In_xGa_{1-x}Sb

InAs/In_xGa_{1-x}Sb can be both qualitatively and quantitatively described as a Krönig-Penney potential, or "superlattice".

To understand the properties of a superlattice, it is first necessary to understand those of a single quantum well. An electron in a potential field $U(x)$, where $U(x) = 0$ for $-a/2 < x < a/2$ and $U(x) = U_0$ everywhere else, can be said to exist in a quantum well, or a finite square well (see Fig.1). It is necessary to solve the Schrödinger wave equation to locate the allowed positions of the electron within the well.

The time independent Schrödinger equation reads:

$$\frac{-\hbar^2 d^2 \Psi(x)}{2\mu dx^2} + U(x)\Psi(x) = E\Psi(x)$$

where μ is the mass of the electron, \hbar is Heisenberg's constant over 2π , $\Psi(x)$ is the wave function of the electron, and E is the energy of the electron. The solutions are simple:

$$Ae^{ikx} + Be^{-ikx}$$

for $-a/2 < x < a/2$ and

$$Ce^{-iKx} + De^{iKx}$$

everywhere else. A, B, C , and D are constants that are chosen subject to the normalization condition, such that the area underneath the square of the wave function must equal one. k and K equal, respectively:

$$\frac{\sqrt{2\mu E}}{\hbar^2} \quad \frac{\sqrt{2\mu(U_0 - E)}}{\hbar^2}$$

By matching the two equations and their derivatives at $x = -a/2$ and $x = a/2$, the secular equation for the allowed energy states of a quantum well can be obtained⁴:

$$\left(\varepsilon \tan \varepsilon - \sqrt{\frac{\mu U_0 a^2}{2\hbar^2} - \varepsilon^2} \right) \left(\varepsilon \cot \varepsilon + \sqrt{\frac{\mu U_0 a^2}{2\hbar^2} - \varepsilon^2} \right) = 0$$

where

$$\varepsilon = \sqrt{\frac{\mu E a^2}{2\hbar^2}}$$

This equation (as well as the remaining equations) is transcendental and cannot be solved algebraically.

A short listing of terminology may be helpful. An electron with $E < U_0$ is said to be in the "well" when it is located between $-a/2$ and $a/2$. This well has, of course, width a and depth U_0 . The electron is said to be in the "barrier" everywhere else. This is the classically forbidden region since a classical particle would not have enough energy to penetrate the barrier. Only electrons of certain energy are allowed within the quantum well. These energies are known as "eigenstates".

A superlattice, then, is a structure made up of quantum wells placed in relatively close proximity to each other. If the wells are placed close enough, the wave function of an electron in one well will extend through the barrier into the next. The electron can therefore "tunnel" through the barrier into the next well, and eigenstates are no longer confined to a single value but spread out over mini-bands² (see Fig. 2).

Since the wave function of the electron propagates throughout the whole superlattice, a slightly different approach must be taken to find the allowed eigenstates. A simple equation can be found for the periodic superlattice, in which all wells and barriers have widths a and b respectively. Since the algebra is rather tedious, only the final result is given here.

$$\cos(qd) = \cosh(Ka)\cos(kb) - \frac{\varepsilon}{2} \sinh(Ka)\sin(kb)$$

Here d is the period of the superlattice, $a+b$. q is the wave vector and is roughly equivalent to momentum, and ε here is $K/k - k/K$. It is easy to see that

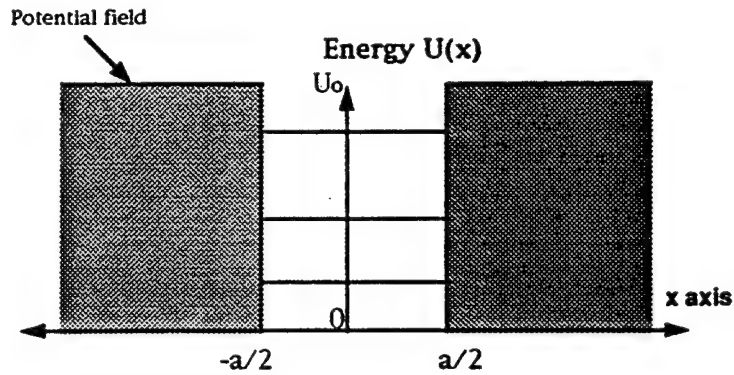


Fig.1 Quantum Well Allowed eigenstates for an electron confined to the well are shown by the horizontal lines within the well. These represent the energy of the electron, not the shape of the wave function. The shaded area represents the classically forbidden region.



Fig. 2 Superlattice (Krönig-Penney potential)
A superlattice is nothing more than quantum wells placed in close proximity. The fact that electrons can tunnel through the small barriers allows the energy states to spread into mini-bands, shown by the horizontal gray areas.

the allowed energies (energy is hidden within K, k , and ϵ) form a band of states because the $\cos(qd)$ can take on any value from -1 to +1 inclusively^{2,3}.

Of course no such simple equation exists for an aperiodic superlattice. However, the transfer matrix method will allow determination of the energy states. With some differences in notation and with the addition of weighted derivatives, the treatment here follows that of Merzbacher³.

Consider a single barrier opposing an otherwise free electron. The wave equation of the electron on one side of the barrier can be related to the wave equation of the electron on the other side through matrices. Two requirements must be imposed. First, the wave equation of the electron must be continuous across the barrier (at $x = \pm b/2$). Second, the derivatives of the wave equation, multiplied by the inverse mass of the electron, must be continuous across the barrier (the electron mass weighting the equation is a result of the envelope function approximation used when dealing with the Krönig-Penney potential). The equations seen previously as solutions to the Schrödinger equation become:

$$\begin{pmatrix} e^{-ikb} & e^{ikb} \\ e^{-ikb} & -e^{ikb} \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} e^{Kb} & e^{-Kb} \\ Ze^{Kb} & -Ze^{-Kb} \end{pmatrix} \begin{pmatrix} C \\ D \end{pmatrix}$$

In a similar fashion the coefficients of the wave equation in the barrier can be related to the coefficients of the wave equation after the barrier.

$$\begin{pmatrix} e^{-Kb} & e^{Kb} \\ e^{-Kb} & e^{Kb} \end{pmatrix} \begin{pmatrix} C \\ D \end{pmatrix} = \begin{pmatrix} e^{ika} & e^{-ika} \\ \Omega e^{ika} & -\Omega e^{-ika} \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix}$$

Here Z and Ω equal, respectively:

$$\frac{iKm_1}{km_2} \quad \frac{ikm_2}{Km_1}$$

with m_1 and m_2 as the masses of the electron outside and inside the barrier respectively.

With a little algebra, the coefficients of the wave equation before and after a barrier of width b can be related:

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} (\cosh Kb + \frac{i\varepsilon}{2} \sinh Kb)e^{ib} & \frac{i\eta}{2} \sinh Kb \\ -\frac{i\eta}{2} \sinh Kb & (\cosh Kb - \frac{i\varepsilon}{2} \sinh Kb)e^{-ib} \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix}$$

This matrix can be written:

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} \alpha + i\beta_1 & i\beta_2 \\ -i\beta_2 & \alpha - i\beta_1 \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix}$$

In the case of a superlattice, consisting of periodic barriers, the coefficients of the wave equation in one well and the next can be related using the notation of the above matrix, with l as the period of the lattice, or $a+b$.

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} (\alpha - i\beta_1)e^{ikl} & -i\beta_2 e^{ikl} \\ i\beta_2 e^{-ikl} & (\alpha + i\beta_1)e^{-ikl} \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix}$$

This is known as the transfer matrix.

Thus, the coefficients of the wave equation in one well can be related to the next through the transfer matrix P .

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = P \begin{pmatrix} A_2 \\ B_2 \end{pmatrix}$$

This process can be extended indefinitely:

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = P_1 P_2 \dots P_{n-1} P_n \begin{pmatrix} A_{n+1} \\ B_{n+1} \end{pmatrix}$$

One last condition must be imposed: the boundary condition. This denotes the boundary of the superlattice, and several possible conditions exist. For instance, the superlattice could be contained within a uniform potential extending to infinity. The boundary condition chosen in this project is known as the periodic boundary condition. This specifies that the superlattice loops back upon itself, so that after n periods, the lattice repeats and continues to repeat to infinity (see Fig. 3). This condition has the advantage of simplifying

calculations. Wave equations that do not meet this condition (do not repeat after n periods) are not solutions of the superlattice.

Given this boundary condition, the above equation can be rewritten as:

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = M \begin{pmatrix} A_1 \\ B_1 \end{pmatrix}$$

with

$$M = \prod_{i=1}^n P_i$$

Thus, energy states exist when:

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} (M - I) = 0$$

where I is the identity matrix.

InAs/ $\text{In}_x\text{Ga}_{1-x}\text{Sb}$ is constructed, using molecular beam epitaxy, from alternating layers of InAs and $\text{In}_x\text{Ga}_{1-x}\text{Sb}$, which constrain electrons in the manner of the Krönig-Penney potential. However, this model must be extended to explain the electro-optical properties of InAs/ $\text{In}_x\text{Ga}_{1-x}\text{Sb}$. In particular, the two-band model serves as a useful approximation.

Upon absorbing a photon, a valence electron will attain an excited state and conduct throughout the InAs/ $\text{In}_x\text{Ga}_{1-x}\text{Sb}$ superlattice. Once the electron is detected at a contact, it is obvious that a photon has hit the InAs/ $\text{In}_x\text{Ga}_{1-x}\text{Sb}$ superlattice (hence its application in infrared detectors). Upon moving to an excited state, the electron will leave behind a "heavy hole", a region of positive charge that behaves according to all the equations previously given. While InAs serves as the well for the electron, the $\text{In}_x\text{Ga}_{1-x}\text{Sb}$ serves as the well for the heavy hole. This phenomenon, whereby the well of the electron becomes the barrier for the heavy hole, is known as Type 2 stagger. The band diagram for InAs/ $\text{In}_x\text{Ga}_{1-x}\text{Sb}$ can be drawn as in Fig. 4. Thus, this model contains two bands, the valence band for heavy holes and the conduction band for excited electrons.

For an electron to absorb a photon, two conditions must be met. First,

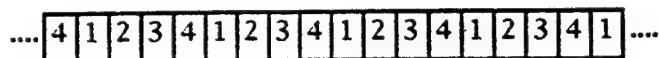


Fig. 3 Periodic Boundary Condition Each block represents one period - one well and one barrier - of the superlattice. After n periods, in this case 4, the superlattice repeats.

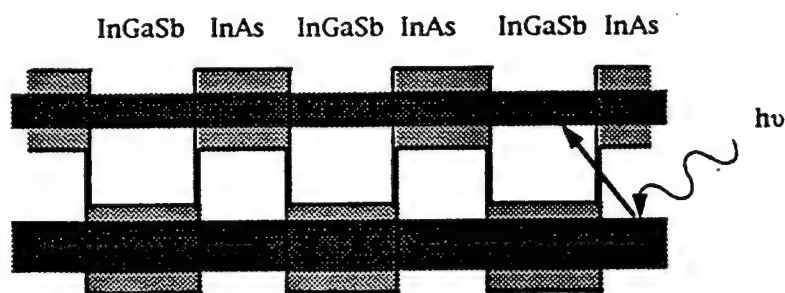


Fig. 4 InAs/In_xGa_{1-x}Sb Band Diagram
Valence band is at the top, and conduction band is at the bottom. InAs/In_xGa_{1-x}Sb displays Type II stagger. A photon is absorbed by an electron in the valence band, boosting the electron to the conduction band.

the energy of the excited electron must lie somewhere within an allowed miniband. Second, the heavy hole must also lie within a miniband. If either of these conditions are not met, the photon cannot be absorbed.

By taking the difference between the lowest energy of the conduction band and the highest energy of the valence band, one can obtain the band gap for InAs/In_xGa_{1-x}Sb. It is obvious that this bandgap represents the absolute minimum amount of energy that could be absorbed by any valence electron in InAs/In_xGa_{1-x}Sb.

Using the band gap, one can easily obtain the minimum wavelength of a photon that could be absorbed by a detector employing InAs/In_xGa_{1-x}Sb. This minimum wavelength is known as the cut-off wavelength, and along with the bandgap is the focus of this investigation.

III. Methodology

A few parameters must be taken into consideration before a model can be developed from the above equations. First, parameters such the depth of the valence and conduction bands, as well as the effective mass of the heavy hole in the well and the barrier, are given by Heller, *et al.*¹ The effective mass of the electron in the conduction band is given by Heller, *et al.*⁵ in an unpublished work, and is a function of energy:

$$M_c(z) = \frac{3 \left[\frac{\Delta E^2}{2} - (V_{LH}(z) - E)(V_{SO}(z) - E) \right]}{2\Pi^2 [2(V_{SO}(z) - \Delta E - E) + V_{LH}(z) - E]}$$

ΔE is a term that relates to the strain in the superlattice. To simplify calculations, this term is considered to be 0, an approximation that works best when the mole fraction Indium in In_xGa_{1-x}Sb is 0. Further approximations include $V_{LH}=V_{HH}$ and $V_{SO}=E_{SO}$. Π is the Kane parameter. All values are provided by Heller, *et al.*,¹ with linear interpolation by the Vegard Rule if needed.

Units must also be considered. By taking energy in units of Rydbergs ($13.6058 \text{ eV} = 1 \text{ R}$), widths in units of Bohr radii ($.529177 \text{ Angstroms} = 1 \text{ b.r.}$), the free electron mass as $1/2$, and lastly \hbar as 1, all the units will automatically cancel in the calculations.

Armed with both numbers and equations, one can easily develop an algorithm to find the minibands in the conduction and valence bands. All that remains is the development of an efficient numerical procedure, a relatively difficult task.

Given the last equation introduced, allowed energy states occur when $\det|M-I|=0$. It is possible to step through the possible energies, determine the transfer matrix for each well/barrier pair, calculate M , subtract the identity matrix, and then take the determinant. The $\det|M-I|$ can therefore be seen as a function $f(x)$, where x is energy in Rydbergs. Anytime $f(x)$ changes in sign, a root has occurred (this neglects the possibility of degenerate roots, which will be considered later). Knowing the approximate location of the root, regular falsi can be used to iteratively improve the location of the root to virtually any desired accuracy.

It is interesting to note that this equation will not generate mini-bands, but instead, as a result of the periodic boundary condition, will produce a certain number of roots corresponding (with double, quadruple, or even greater degeneracy at times) to the number of transfer matrices used. All the roots will be either at the boundary of the actual miniband or inside it. As the number of transfer matrices used approaches infinity, the roots will approach the true miniband.

The first difficulty lies in the fact that matrix $|M-I|$ is extremely ill-conditioned. Ill-conditioning is a term used in numerical analysis for a matrix which, given an extremely slight change in any element, has an extremely large change in the final result of matrix operations, in this case the taking of the determinant⁶. Numerical calculations involving ill-conditioned matrices require extraordinary amounts of computer precision so that round-off error will not cause the determinant to swing wildly (The subtraction of the identity

matrix and the subtraction involved in finding the determinant are the main causes of the round-off error, but the large number of calculations needed to find matrix M must also be taken into account.).

The ill-conditioning of matrix $|M-II|$ can cause extremely large changes in $f(x)$. This problem leads to the detection, and iterative improvement, of false roots.

The best method to deal with an ill-conditioned matrix is to find another method. After some experimentation, the following method was developed.

After writing out the terms in $\det|M-II|$, one obtains $\det|M| - \text{tr}|M| + 1 = 0$. Since the determinant of any of the transfer matrices is 1, and the product of the determinants is the determinant of the products, the determinant of M is also 1. Therefore, this reduces to: $2 - \text{tr}|M| = 0$. The trace of $|M|$ is always real, as shown by the following argument.

Any transfer matrix P can be written as:

$$\begin{pmatrix} P_{1,1} & P_{1,2} \\ P_{1,2}^* & P_{1,1}^* \end{pmatrix}$$

where * denotes complex conjugate.

It is a trivial matter to show that the product of any number of matrices of this form is another matrix of this form. Therefore, the $\text{tr}|M|$ reduces to $2\text{Re}(M_{1,1})$.

Using the trace of $|M|$ rather than the determinant of $|M-II|$ solves the problem of false roots, but leaves another obstacle: overflow. The following method appears to work best.

By inspection, it is obvious that the COSH and SINH terms cause the overflow. We can factor each transfer matrix p like so:

$$P_i = \cosh(2K_i a_i) \begin{pmatrix} (\alpha_1 - i\beta_1)e^{ikl} & -i\beta_2 \\ i\beta_2 & (\alpha_1 + i\beta_1)e^{-ikl} \end{pmatrix}$$

$$\alpha_1 + i\beta_1 = \left(1 + \frac{i\varepsilon}{2} \tanh 2Ka\right) e^{2ika}$$

$$\beta_2 = \frac{-i\eta}{2} \tanh 2Ka$$

If the above matrix is designated q , then:

$$\text{Tr}(M) = \left[\prod_{i=1}^n \cosh(2K_i a_i) \right] \left[\text{Tr} \prod_{i=1}^n q_i \right]$$

Since solutions occur for $2 = \text{Tr}|M|$,

$$\ln 2 = \sum_{i=1}^n \ln[\cosh(2K_i a_i)] + \ln \left[\text{Tr} \prod_{i=1}^n q_i \right]$$

Defining the product of all the q -matrices to be a new matrix $|M|$,

$$-\sum_{i=1}^n \ln[\cosh(2K_i a_i)] - \ln \text{Re}(M_{1,1}) = 0$$

which is the equation used in this model. The left hand side is a function of energy and will be referred to as $f(x)$.

While controlling overflow, this equation presents its own difficulties. The real part of $M_{1,1}$ can easily go negative. Thus, it appears that roots must be found by searching through complex space (where the logarithm is multi-valued, a further difficulty to be considered).

However, this problem can be avoided through an interesting manner. Instead of using n periods to find matrix $|M|$, the use of $2n$ periods, with the second half identical to the first half, has resolved the problem in all cases

seen so far. In effect, with the trace of $|M|$ negative, the trace of $|M|^*|M|$ is positive. The mathematical basis for this is not yet understood.

This leaves the case $\text{Re}(M_{1,1}) = 0$. In all cases seen so far, $f(x)$ will be negative at the bottom of the well (energy=0). Thus for $\text{Re}(M_{1,1}) = 0$, $f(x)$ equals $+\infty$, and a root has occurred between $f(x) < 0$ and that point. The detection of the root is then fairly simple.

As $\text{Re}(M_{1,1})$ approaches 0, $f(x)$ could attain values large enough to cause overflow. However, this has never been observed.

The possibility of degenerate roots must be considered (for degenerate roots $f(x)$ will not cross the x-axis and so will not be detected by the normal sign-change routine). With $f(x)$ beginning negative, a degenerate root will be a local maximum and can thus easily be found. It was discovered that $f(x)$ is so smooth that any local maximum can be taken as the location of a root, because what has the appearance of a local maximum is in fact an asymptote.

The last major problem remaining is speed. While the calculation of $f(x)$ for any given energy is not prohibitive, the actual shape of the function can only be seen at resolutions of $1E-7$ or greater. Thus, starting at the bottom of the well and increasing x by steps of $1E-7$ ($1E-9$ is to be preferred by $1E-7$ is perfectly acceptable), requires a few hours of computation time.

An intelligent guessing routine increases the speed enormously. Any given well/barrier pair $a+b$ has a corresponding lowest allowed energy state for a periodic superlattice with period $a+b$. In the case of an aperiodic lattice, the minibands can be seen as the superposition or average of all the minibands corresponding to the period case of each individual well/barrier pair. This allows the program to virtually pinpoint the location of the miniband.

However, even with this increase in speed, locating the conduction band requires approximately 15 minutes, which is not as fast as would be preferred.

The final consideration for the model is the type of random number generator (RNG), and how to implement it.

The RNG implemented used the Parks-Miller (revised) equation, known as the minimal standard RNG⁸. In order to transform the uniform deviates from the P-M equation into a normal distribution, the well-known Box-Muller equation must be used:

$$z = [\sqrt{-2\ln(u)}]\sin(2\pi v)$$

with u and v as uniform deviates.

z is then multiplied by the desired standard deviation, which is 3 Angstroms for this model, this being the approximate size of one layer of atoms in the $\text{InAs}/\text{In}_x\text{Ga}_{1-x}\text{Sb}$.

Some belief exists that while the well and barrier widths of a layered material like $\text{InAs}/\text{In}_x\text{Ga}_{1-x}\text{Sb}$ can vary during the growth process, the period remains roughly the same⁷. In other words, for each period with well width a and barrier width b , $a+b$ is a constant. For this investigation, the period was kept constant in one trial and was allowed to vary in the other.

This finalizes the model used in the investigation.

IV. Results

In the first of two trials, the period of the superlattice is kept constant to 50 angstroms. In both trials, the mole fraction Indium is 0, to obtain the best accuracy with the approximate model. The InAs layer has a mean of 30 angstroms, while the GaSb layer has a mean width of 20 angstroms. The standard deviation of the change in width is 3 angstroms, corresponding to 1 monolayer of Indium and Arsenic atoms. The results are plotted in the first graph. For comparison, three band gaps for periodic cases are listed on the graph.

The band gap is generally decreased due to the variation in layer widths, and only a very few points have a band gap above that of the mean width (30/20). Most points are contained within what appears to be a close to normal distribution centered at the average band gap of .26 eV. The cut-off wavelength varies relatively little, and most points are contained between 4.2 and 5.6 μm . The data shown in the graph represents 95 points.

The second trial allows the InAs and GaSb layers to vary independently, again with a standard deviation of 3 angstroms. However, the mean width of the InAs layer is 62 angstroms, and the mean width of the GaSb layer is 22 angstroms, which is near the optimum absorption for this model. The results are summarized in the second graph, containing 399 points.

Once again the band gap takes on relatively uniform distribution, between .147 and .092 eV, centered at .120 eV, which is close to the average of .116 eV. A few points lie a considerable distance from the rest, including one point at .177 eV and one at .079 eV. The cut-off wavelength, which is 10 μm for the mean case, fluctuates between 8.3 and 13.5 μm . The two outliers just mentioned have cut-off wavelengths of 7.0 and 15.7 μm respectively.

V. Conclusion

Random variations in the layer widths of $\text{InAs}/\text{In}_x\text{Ga}_{1-x}\text{Sb}$ generally decrease the size of the band gap and increase the cut-off wavelength. If the period is held constant, the decrease in band gap occurs almost exclusively, but for freely varying layer widths, the band gap can increase in size with an almost equal probability. The cut-off wavelength is generally increased but for freely varying periods can also be decreased.

For some uses, a cut-off wavelength that varies $\pm 2 \mu\text{m}$ is acceptable, but the results indicate that control of 1 monolayer or less over the growth of $\text{InAs}/\text{In}_x\text{Ga}_{1-x}\text{Sb}$ can cause fluctuations in the cut-off wavelength that exceeds even $\pm 2 \mu\text{m}$. In cases where greater uniform detector response is required, the growth must be controlled to more than 1 monolayer.

Naturally further study is needed, both to more accurately judge the shape of the curve and to predict the probability that extreme outliers will occur.

VI. Acknowledgements

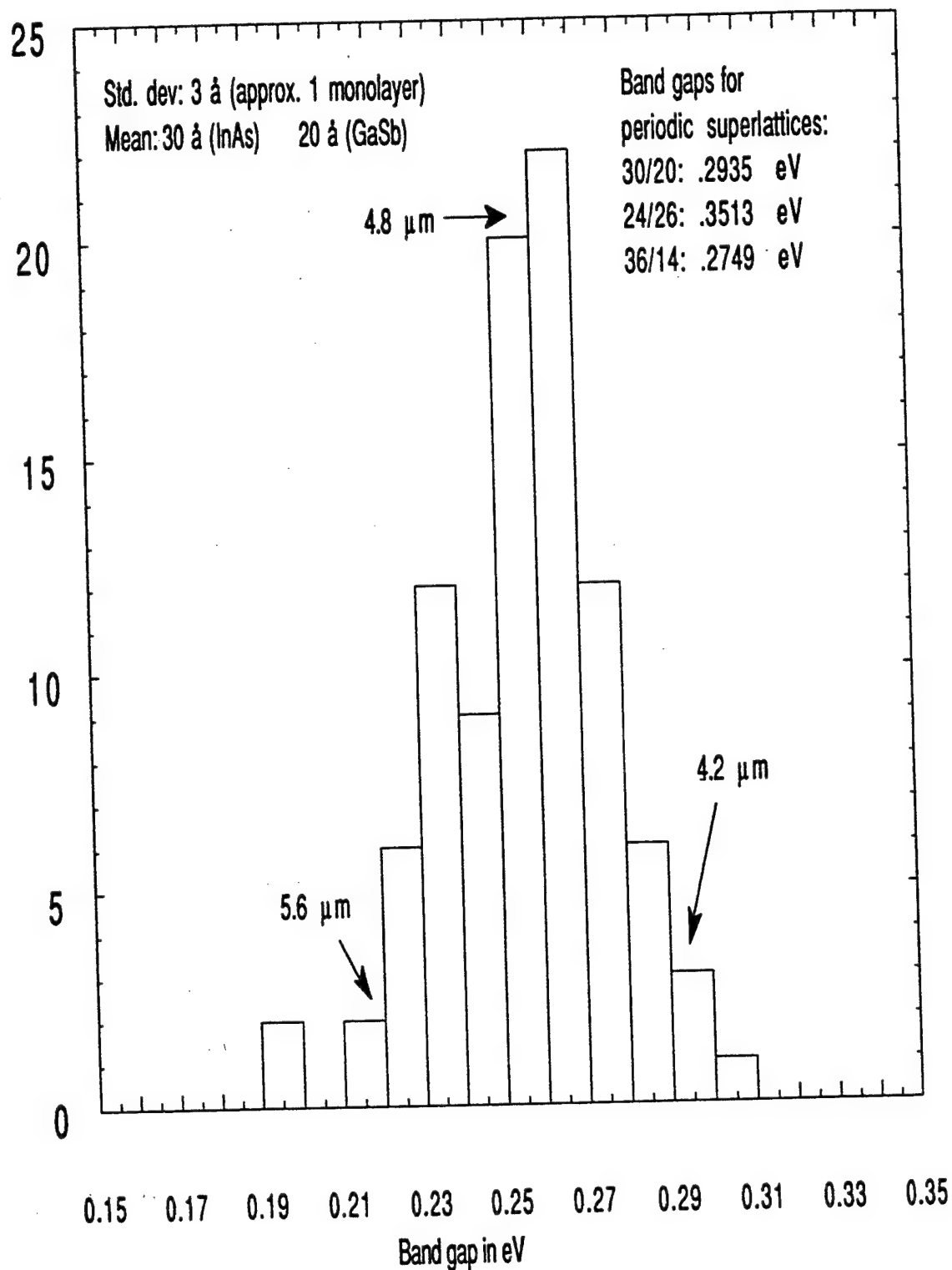
The author would like to gratefully acknowledge Dr. Pat Hemenger for sponsoring this project, Dr. Frank Szmulowicz for his advice and encouragement, and Eric Heller for much helpful assistance. The author would also like to acknowledge AFOSR and Wright Labs for the opportunity to undertake this project.

VII. References

- ¹E.R. Heller, K. Fisher, F. Szmulowicz, and F.L. Madarasz, J. Appl. Phys. 77, 5739 (1995).
- ²M. Jaros, *Physics and Applications of Semiconductor Microstructures* (Oxford, New York, 1989).
- ³E. Merzbacher, *Quantum Mechanics* (Wiley, New York, 1970).
- ⁴R.M. Eisberg, *Fundamentals of Modern Physics* (Wiley, New York, 1961).
- ⁵E.R. Heller, K. Fisher, F. Szmulowicz, and F.L. Madarasz, "Optimization of Absorption in InAs/In_xGa_{1-x}Sb Superlattices for Long Wavelength Infrared Detection" (unpublished).
- ⁶See, for example, A. Ralston, *A First Course in Numerical Analysis* (McGraw-Hill, New York, 1965).
- ⁷See, for example, M. Lakrimi, T.A. Vaughan, R.J. Nicholas, N.J. Mason, and P.J. Walker, "Long Wavelength Photoresponse of Short Period InAs/GaSb Superlattices".
- ⁸S.K. Park, K.W. Miller, "Random Number Generators: Good Ones Are Hard to Find", *Computing Practices* 31, 1192.

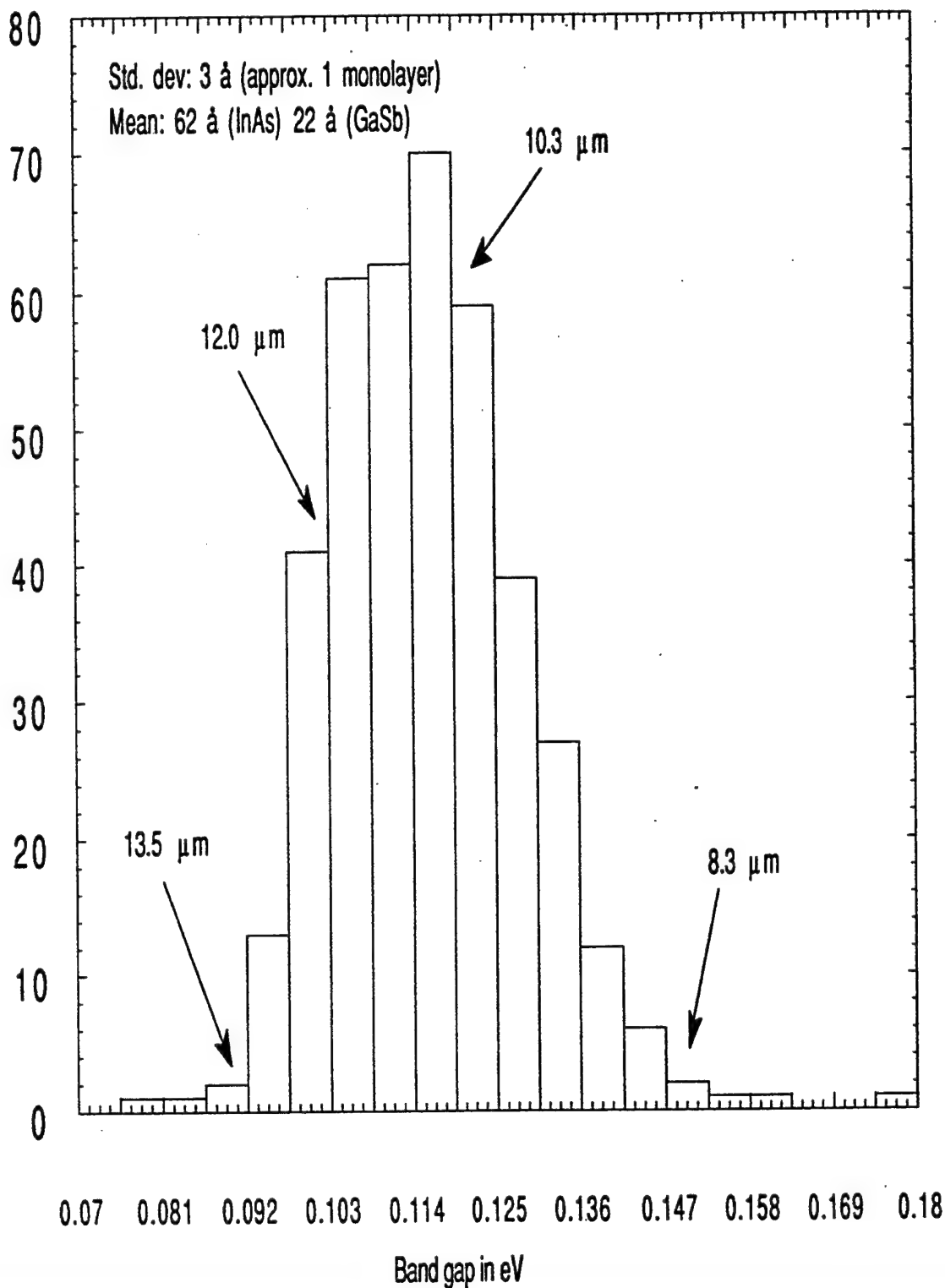
Band Gaps for Random Well/Barrier Widths of InAs/GaSb With Constant Period

Number of band gaps within the specified range



Band Gaps for Random Well/Barrier Widths of InAs/GaSb With Varying Period

Number of band gaps within the specified range



Implementation of Customized Database Functions for the Analysis of High-Performance
Electronics

Richard Thompson

Walter E. Stebbins High School
1900 Harshman Rd
Riverside, OH 45424

Final Report for:
High School Apprentice Program
Wright Laboratories

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratories

August 1995

Implementation of Customized Database Functions for the Analysis of High-Performance Electronics

Richard Thompson
Walter E. Stebbins High School

Abstract

Implementing customized database functions for easier access and manipulation of large datasets was the area of development focused on. In researching and developing high-performance electronics, it is often vital that the vast quantities of data collected be readily accessible for analysis of large, dissimilar, complex datasets. Without the ability to quickly retrieve these datasets in acceptable forms that can be analyzed, researchers are often forced to spend many long hours manipulating the collected data into the formats needed for their analysis, creating stagnant development environments encumbered by the lack of information needed to make progress in their research.

Implementation of Customized Database Functions for the Analysis of High-Performance Electronics

Richard Thompson

Introduction

Research and development of high-performance electronics involving gallium-arsenide (GaAs) compounds depends upon the ability to analyze the data collected in order to determine what areas of focus are necessary for the next stage of development. Wafermaps, histograms, trend charts, and correlations between many datasets from various different testing conditions aid researchers in locating the specific areas where problems have occurred or may arise, to arrive at conclusions concerning the testing procedures used, and determine whether or not to continue that avenue of development. A front-end system capable of automating the process of accessing, retrieving, and manipulating the data sets stored within a database into any combination of file formats that may be necessary for the analysis step in research becomes an increasingly more viable alternative to the painstaking manual splicing of data into formats capable of being used for this analysis.

Several standards become useful for the analysis of these datasets. First, the Macintosh, with its superior graphical rendering capabilities, serves as the main platform for the analysis of the collected data. Also, the Macintosh, with its graphical interface, provides an intuitive environment that is easy to use. Second, Oracle serves as the database system mainly because it is "designed to receive large quantities of many different types of data" {1}. This provides the flexibility required for a growing database where new data is always being added in. Lastly, the programming language C has been utilized to develop new applications as it has become an industry standard for programming on the desktop. As an industry standard, many products including Oracle, have supplied libraries containing code for use in producing C applications. With this capability many customized applications could be produced to manipulate the data within Oracle.

Discussion of Problem

In developing and testing high-performance electronics using gallium-arsenide compounds, several steps are taken. First wafers are fabricated and broken into lots according to common processes. Then test devices and "real" transistors are created in reticles. After this, many tests are performed on the devices and transistors to collect as much information as possible, often with multiple results from each test. Then the data collected is analyzed in wafermaps, trend charts, histograms, and many other correlations between the data. The analysis though, may take varying degrees of complexity from comparing the results of individual devices within a single reticle from one wafer, to comparing devices on different reticles, to looking at various properties and trends across many different wafers for any number of tests performed.

The problem arises in placing all the necessary data for analysis into a format that can be used by the various analysis applications. These formats change with differing ranges of complexity from piecing together simple datasets to comparing whole wafers to each other by correlating each piece of data with another. When performed manually, many exhausting hours are spent in retrieving the data required and splicing them to match the test conditions of other sets of data. This method proves to be very tedious and mistake prone, as one wrong placement of data may skew the conclusions into a completely different area of focus than where the problems may lie. So development of a front end system capable of automating this entire process would give researchers more time to manage, assurance of the accuracy of the correlations between the data, and an overall decrease in the time required for analysis of data to be performed .

Developing a front end system which researchers could use to produce data in a specific format for the analysis introduces several problems that must be addressed: how the data is stored in the database, what formats should the data be retrieved in, what formats the data should be loaded in, the type of interfacing required to access the database, and how much information must be supplied by a user to produce the desired formats containing only the information needed. Since clear-cut solutions were not available, research was needed to determine the best methods to use.

Methodology

The first problem addressed was the format in which data would be stored within the Oracle database. Since at the time there were only HBT(heterojunction-bipolar transistor) test data to load, a schema was created within the database to develop several tables each storing different sets of information that could be pieced together in any fashion needed. This schema allows for future indexing of the data stored within the database for greater speed in the retrieval of desired data by incorporating relational queries to block out searches of unnecessary sets of data. As other data becomes available new schemas can then be added to incorporate them into the database.

In loading the data into the database, several problems arose. One, the file format used to store the data was incompatible with the Oracle loader being used, and two, there were too many files to manually load onto the database. When loading the data, each new set of test was required by Oracle have several unique numbers given them so that the test data would be properly stored and not confused with other data later on. So the user loading the test data into the database would constantly have to keep track of the values stored within Oracle to determine what the assign for the new data. Since there were so many files to work with, this became an impossible task to perform manually, and so an automated application was required to handle the file conversion and assignment of various identifying numbers for each test set. Using MetroWerks C, an HBT conversion program was developed to handle the conversion of multiple HBT data files into an Oracle "friendly" format and also capable of accessing Oracles database to retrieve the next set of values to give new test data. With this application all a user was required to

do was to provide the names of the files to convert and the rest of the conversion would be performed while the user went on with other things.

The HBT data conversion application required an additional application to be produced that was capable of accessing the Oracle database through standard SQL (Structure Query Language) statements. SQL is a relational database language used by Oracle designed for use in querying and data manipulation. {2} Using several libraries provided by Oracle for Think C, a shell program was developed which would run in the background to receive Apple Events from other applications wishing to query the database. The reason for the separate application came from a decision to use primarily the MetroWerks C compiler for application development, and that the libraries supplied were only for Think C. To continue to be able to use MetroWerks this shell program was developed. With the development of the conversion application, less time would be spent in preparing data for loading.

Once the data was being loaded, the problem of creating a front end user interface for accessing the database was focused upon. Since researchers did not have the time to fully learn how to use SQL to query a database and piece together the data retrieved, the front end would have to handle the retrieval and splicing of data into user defined formats. The main goals for the front end were to generalize it as much as possible and make it expandable for later changes within the database. The reason for creating a generalized front end was to allow the user to be able to create entire new file formats for the results they would need. In addition, as other test data for other types of transistors are added, expandability was required to handle the new types of data.

The front end user application would be programmed in MetroWerks to take advantage of the ToolBox functions provided by the Macintosh and the SQL shell program already developed. Several features such as allowing the user to create templates of the file formats needed and being capable of producing multiple files each containing varying datasets and tests would be required. Also, since it would be a user interface, it had to be intuitive enough to be easy to use.

Once a working front end system was developed other problems such as addressing the speed problem in having to wait for data retrieval from the database, and the addition of new types of tests would have to be addressed.

Results:

Several new applications were developed to aid in reducing the workload of researchers: an HBT file conversion application capable of converting the test files in a format Oracle could read, an SQL shell application to receive queries from other applications to access the Oracle database, and a front end user application for allowing the data to be manipulated into user defined file formats.

Conclusions:

Overall the research and development of the customized database functions have been an overwhelming success. It has demonstrated that the data held in the Oracle database can

be accessed and manipulated for analysis by researchers, and work is being continued in this area to add even more functionality into the developed customized applications to further reduce the time spent in analysis of data.

References:

1. Hursch, Jack L. & Hursch, Carolyn J. *Working with Oracle version 6.0*, Windcrest Books, 1989. Oracle as a flexible database.
2. Portfolio Tom, *PL/SQL User's Guide and Reference*, Oracle Corporation 1992. SQL description.

Construction of Electrochromic Devices Utilizing Conducting Polypyrrrole

Albert Tu

Niceville Senior High School
800 E. John Sims Parkway
Niceville, FL 32578

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

CONSTRUCTION OF ELECTROCHROMIC DEVICES UTILIZING CONDUCTING POLYPYRROLE

Albert Tu
High School Apprentice
Fuzes Branch
Wright Laboratory Armament Directorate

Abstract

Since their discovery in 1977, conducting polymers have been studied extensively with a special interest in applicational use. The field of polymeric electronic displays has especially drawn attention[1-8]. The purpose of the investigation was to construct a feasible electrochromic device utilizing a thin film conducting polymer. Polypyrrole was electrochemically synthesized with a range of dopants. Thin film polypyrrole was used in the construction of a multi-layered electrochromic device. The electrochromic characteristics were then investigated to create improved devices.

CONSTRUCTION OF ELECTROCHROMIC DEVICES UTILIZING CONDUCTING POLYPYRROLE

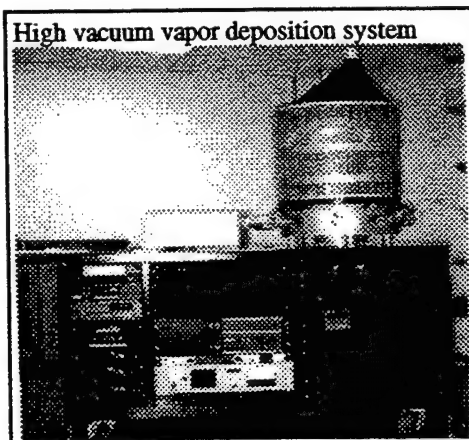
Albert Tu

Introduction

Once it was established that certain polymers can be synthesized to carry electric charge and preselected conductivities could be attained with a certain accuracy, research exploded in the development of feasible applications[9]. Conducting polymers present an especially fruitful application in electronic displays because of 1) low cost relative to current fabrication methods, e.g., liquid crystals, 2) ease of fabrication, and 3) their potential to create displays of large surface area. These displays primarily use conducting polymers as the semiconducting component. Light sensitive devices, 'smart windows', and electrochromic devices are similar to rechargeable batteries in that can be alternated back and forth for long durations[10].

Discussion

Problems confronting the development of polymeric electronic displays involve the interface between conducting polymer and metallic components. Classical methods of soldering render the polymer effectively useless. A new method must be perfected before out of lab fabrication can occur. A promising method is of vacuum vapor deposition in which the metal to be deposited is atomized by one of several methods, electric burst, ion sputtering, laser vaporization. An effort was made to convert a J.A. Woollam Co. high vacuum system, with an ion magnetron capable of sputtering, into a vapor deposition system. A high current feedthrough was installed in anticipation of a low voltage high current power supply to be used to vaporize metals. This system, however, was not on-line for several weeks. An alternative method of metal polymer interface was explored with the use of Nickel print. The print was applied in liquid state and then dried to form a



solid form of nickel. This method appeared satisfactory and did not harm the polymer metal interface. It's efficacy will be compared to a vapor deposited interface as soon as this is possible. Nickel print was used for all devices described in this paper.

Experimentation

From initial research, it was decided that the polymer films used would have to be extremely thin and still uniform. These films would have a preferable thickness of at or below 500 angstroms. To achieve a quality thin film polypyrrole, a wide range of methods of synthesis were explored. Conducting polypyrrole is synthesized by electropolymerization in a solution with the dopant present. Since a direct relationship can be observed between specific dopant to monomer molar concentrations and ratios, specific conductivities can also be achieved with a degree of certainty.

Three dopants were investigated:

Sodium *p*-toluenesulfonate(tosylate), Dodecylbenzenesulfonic acid-sodium salt(DBS)

Lithium TetraFluoroborate (LTF)

Each dopant was tested on a range of molar ratios and concentrations:

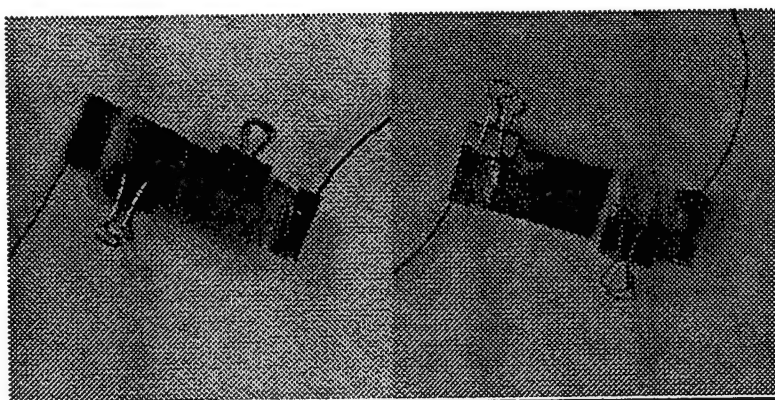
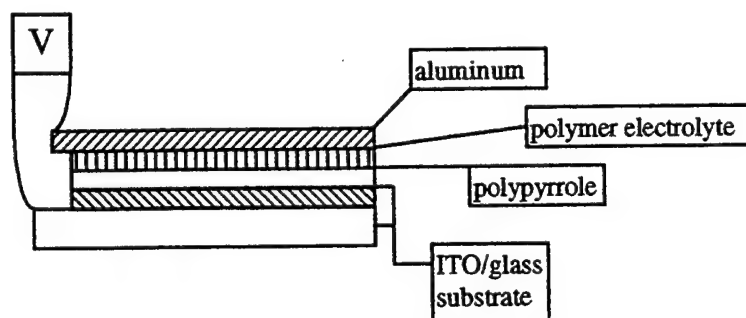
.10M to .15M dopant to .10M to .15M pyrrole

The electric domain tested was:

5volts- 20 volts and 1milliamp/cm² to 5milliamp/cm²

Sodium *p*-toluenesulfonate created films best suited for electrochromic devices. Tosylate doped PPY produced thin uniform films of high quality and greatest consistency. It was found that a formula of .10M pyrrole to .15M tosylate was most efficient. Two methods of electrical force were found to be satisfactory: 1) a high current(5milliamp/cm²), high voltage (20V) burst and 2) a low current(>1milliamp/cm²), low voltage(5V) long duration deposition. The first method is more suitable because of its short duration and greater ease of control.

Multilayered electrochromic devices were constructed using tosylate doped PPY as the semiconducting layer. An Indium Tin Oxide layer that was vapor deposited to a thickness of ~500 angstroms on a glass substrate was used as an electrode. Indium Tin Oxide's property of optical transparency allowed the planned electrochromic behaviour to be observed. Five ITO/glass substrates were provided by NASA's Jet Propulsion Laboratory. The next layer was thin film polypyrrole deposited on the ITO/glass electrode. The next layer was a polymer electrolyte and the final layer was an Aluminum electrode. These components were assembled and then secured with two clamps. A low voltage electrical current was applied and the devices were observed for electrochromic characteristics.



Figures:(clockwise from the top) diagram of multilayered electrochromic device, Device#1-undoped brown/black, Device#2-doped light blue device.

Observations

Eight devices were constructed, one at a time. When an electrical force is applied and the anode is the aluminum electrode, the color of the display changes from brown/black to light blue. This is due to the ion flow from the polymer electrolyte into the PPY doping the polymer film[10]. When the polarity is reversed the PPY was undoped, causing a color change back to brown/black. This phenomenon was observed for all devices constructed, however, as the construction process progressed, the devices achieved a greater visible color change. This could be a result of increased experience in fabrication techniques. Transition time was lowered to less than one second. One limitation was observed. The maximum life cycle of the devices could not be raised over ~10. That is, the maximum number of doping/undoping cycles was ten. This could be due to the long term effect of the electrical force on the PPY[11,12].

Conclusion

Electrochromic devices were constructed utilizing conducting polypyrrole and a definite doping and undoping process was observed. The electrochromic characteristics were observed to be satisfactory. One drawback was noted, the limited life span of the devices. This, however, could be due to the fabrication process. The efficacy and efficiency of the devices constructed will be compared to devices constructed with vapor deposited polymer/metal interfaces to determine if the nickel print interface is the source of the problem.

Acknowledgments

Thanks to Dr. Duane Finello, mentor, MNSE for the use of the chemistry lab, Ron Boulet, Fuzes Branch Chief, and Don Harrison and Mike Deiler, program co-coordinators.

References

1. I.D. Parker, "Carrier tunneling and device characteristics in polymer light-emitting diodes," *J. Appl. Phys.* 75 (3), 1 February 1994.
2. V.L. Colvin, M.C. Schlamp & A.P. Allvisatos, "Light-emitting diodes made from cadmium selenide nanocrystals and a semiconducting polymer," *Nature* 370 4 August 1994.
3. Marko Strukelj, Fotis Papadimitrakopoulos, Timothy M. Miller, Lewis J. Rothberg, "Design and Application of Electron-Transporting Organic Materials," *Science* 267 31 March 1995.
4. Minoru Umeda and Mitsuru Hashimoto, "Study of photocarrier generation mechanism in a layered photoreceptor: Triphenylamine trisazo pigment/ molecularly doped polymer,"
5. J. Kido, et al. "White light-emitting organic electroluminescent devices using the poly(N-vinylcarbazole) emitter layer doped with three fluorescent dyes." *Appl. Phys. Lett.* 64 14 February 1994.
6. J. Kido, et al. "Bright red light-emitting organic electroluminescent devices having a europium complex as an emitter" *Appl. Phys. Lett.* 65 24 October 1994.
7. M. Berggren, et al. "Light emitting diodes with variable colours from polymer blends." *Nature* 372 1 December 1994.

8. G. Phillips, et al. "Dielectric properties of polypyrrole doped with tosylate anion in the far infrared and microwave." *J. Appl. Phys.* 69 15 January 1991.
9. M. Aldissi Inherently Conducting Polymers. Noyes Data Corporation. 1993.
10. H. Naarman, "Science and Applications of Conducting Polymers", IOP Publishing Ltd., 1991.
11. L.M. Do, et al. "Observation of degradation processes of Al electrodes in organic electroluminescence devices by electroluminescence microscopy, atomic force microscopy, scanning electron microscopy, and Auger electron spectroscopy." *J. Appl. Phys.* 76 1 November 1994.
12. Syun Egusa, et al. "Carrier injection characteristics of the metal/organic junctions of organic thin film devices" *J. Appl. Phys.* 71 15 February 1992.

CHARACTERIZATION OF OPTICAL FILTERS BUILT USING SYNTHETIC IMAGERY:
PHASE II

John W. Vest

Niceville Senior High School
800 E. John C. Sims Parkway
Niceville, Fl 32578

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

CHARACTERIZATION OF OPTICAL FILTERS BUILT USING SYNTHETIC IMAGERY: PHASE II

**John W. Vest
Niceville Senior High School**

Abstract

The current method of image generation for optical filter design is to build scale models of targets which are then photographed at the desired angles and then digitized. This method is both costly and time consuming. It was hypothesized that a battlefield scene modelling program called Irma could be used to create imagery that could be used instead of the scale models. A method of image creation was developed and filters were made from the Irma images. Test results and comparisons to filters built using the scale model images show that this is indeed a better method of image production.

CHARACTERIZATION OF OPTICAL FILTERS BUILT USING SYNTHETIC IMAGERY: PHASE II

John W. Vest

Introduction

The United States Air Force is developing optical filtering techniques for use in advanced guidance seeker technology. The main use for this pattern recognition technology is for the implementation of target identification hardware on advanced seeker munitions. To make this effective, a bank of filters must be stored on the munition to provide it with a wide range of identification possibilities. To make these filters, images sets of many different targets must be obtained. Currently, the main method of target image creation is to build scale models of desired targets, photograph the models at specified angles, and digitize the imagery. This method is extremely time consuming and not very cost efficient. Once the models are purchased and assembled, the tedious job of photographing the target in one degree rotation increments can take an extremely long time, yielding results that are not always up to standard. Therefore, a new, improved method of image generation must be developed to insure accurate and speedy images sets for optical filter design. The most promising solution to this need is found in computer generated synthetic imagery.

Nichols Research Corporation, under contract to the United States Air Force Wright Laboratory/Armament Directorate Air-to-Surface Guidance Branch, has developed a software package known as Irma. Irma, originally an acronym for InfraRed Modeling and Analysis, "is a computer simulation that generates accurate synthetic imagery of typical battlefield targets and backgrounds as observed by passive sensors." It was hypothesized that this program could be used to create sequences of images capable of being used for optical filter generation. The move from scale models to facet based computer generated models will prove to be a major advance in optical filter development.

Methodology

I. Image Synthesis

The first step in developing this new method of optical filter design was to learn to create images using the Irma modeling codes. Phase I of this project, conducted during the summer of 1994 for the High School Apprenticeship Program, dealt directly with this critical move. The results from those initial experiments with the Irma simulation resulted in a target set of IR T62 images. Figure 1 is an example of these images, this particular image taken at a 90° rotation and against a black background. Unfortunately preliminary tests on optical filters built using these images were a complete failure. The existing images used to create and to test filters are all visible video images. The synthetic images that were created in the IR spectrum, simply did not match to those detected by a video camera operating in the visible spectrum. Therefore, more realistic visible images had to be created by Irma for this research to ever prove successful.

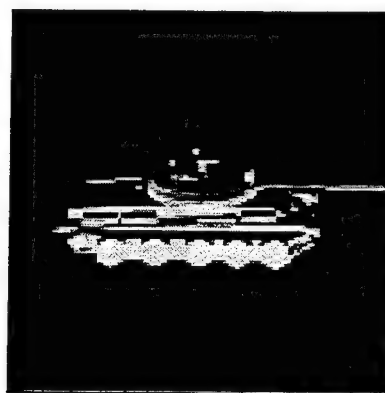


Figure 1
IR T62

The Irma modeling software, as evidenced by its name, was primarily designed to simulate an infrared detector. Improvements in the software have added capabilities such as active/passive infrared, active/passive millimeter wave, and LADAR image generation. Unfortunately, the simulation is really not designed for visible image generation. The millimeter wave rendering tools being developed are capable of creating visible images, but not of the quality required. Therefore, the passive infrared simulation had to be configured to produce realistic visible images.

Irma is basically comprised of three main programs that use many associated data files for support and input. The two most important of these programs are the ENVIRO model and the Scene and Sequence Generator (SSG). The ENVIRO model determines the physical temperatures of the objects included in a particular simulation. The SSG actually generates the passive signature of the scene as observed by a desired sensor. In creating visible images, no changes were necessary in the execution of the ENVIRO model. The ENVIRO outputs required are the same regardless of what type of imagery is being created. The inputs of the SSG, however, needed to be altered in order to get the desired results.

The first input file that needed to be changed was the Sensor Spectral Response File, which defines the sensor waveband for the simulation. The files delivered with the Irma software only simulate the responses of infrared and millimeter wave cameras. Therefore, a file needed to be created that can simulate a visible camera. An existing file was found that supposedly did just this. Figure 2 is this Sensor Spectral Response File. The first number on each line is the wavelength for the sensor response and the second number is the sensor spectral response at that wavelength.

0.400	0.0
0.425	0.0073
0.45	0.0380
0.475	0.1128
0.5	0.3230
0.525	0.7832
0.55	0.9950
0.575	0.9154
0.6	0.6310
0.625	0.3210
0.65	0.107
0.675	0.0232
0.7	0.0

Figure 2
Visible Sensor Spectral Response .

The next file that needed to be modified was the Sun File, which describes the effects of the sun on a scene. For generation of passive imagery, this file is optional. Visible images, though, need this file to correctly render the scene. Figure 3 is the Sun File used in these experiments. After the first two lines, the first number on a line is a wavelength and the second number is the solar irradiance at that wavelength.

0.4 -0.5 0.8	.515 0.084
48	.526 0.0794
.350 0.01242	.538 0.0914
.355 0.01744	.549 0.096
.360 0.01898	.562 0.0956
.365 0.01974	.575 0.099
.370 0.01992	.588 0.0968
.376 0.0218	.602 0.0994
.382 0.0206	.617 0.1016
.388 0.0202	.633 0.1036
.394 0.01224	.649 0.101
.400 0.0382	.667 0.1074
.407 0.0396	.685 0.1068
.413 0.0452	.704 0.104
.420 0.0466	.725 0.0914
.427 0.0448	.746 0.1038
.435 0.049	.769 0.098
.442 0.0612	.794 0.0968
.450 0.0692	.820 0.0858
.459 0.0666	.847 0.0926
.467 0.0706	.877 0.0936
.476 0.0726	.909 0.0664
.485 0.0712	.943 0.038
.495 0.0808	.980 0.0754
.505 0.0818	1.000 0.0816

Figure 3
Sun File

Another file that was not necessary for the generation of infrared images is the Background File. This file contains the skyshine and earthshine radiance information. The Background File is optional, although it was used to generate the visible images. Figure 4 is the Background File used in these experiments. The numbers describe background radiances at specified wavelengths.

63 9	0.592000 0.0686	0.420000 0.0814	0.870000 0.03772	0.543000 0.0796571
0.000	0.602000 0.068	0.426000 0.0802286	0.893000 0.03412	0.552000 0.0778857
0.350000 0.033	0.613000 0.0666	0.431000 0.0671143	0.917000 0.0243829	0.562000 0.0764571
0.353000 0.0424	0.625000 0.0646	0.437000 0.0922	0.943000 0.0118943	0.571000 0.0749429
0.357000 0.0372	0.637000 0.0648	0.442000 0.0947429	0.971000 0.0235257	0.581000 0.0764857
0.361000 0.0348	0.649000 0.0596	0.448000 0.0952	1.000000 0.0244314	0.592000 0.0706571
0.365000 0.046	0.662000 0.0606	0.455000 0.0954286	25.000	0.602000 0.0703143
0.369000 0.05	0.676000 0.0586	0.461000 0.0983714	0.350000 0.0324857	0.613000 0.0687714
0.373000 0.0366	0.690000 0.0488	0.467000 0.0962286	0.353000 0.0417714	0.625000 0.0669143
0.377000 0.0518	0.704000 0.0518	0.474000 0.0972	0.357000 0.0367429	0.637000 0.0671143
0.382000 0.0426	0.719000 0.0388	0.481000 0.0987143	0.361000 0.0343714	0.649000 0.0616286
0.386000 0.0426	0.735000 0.0468	0.488000 0.0902	0.365000 0.0455143	0.662000 0.0629714
0.391000 0.0618	0.752000 0.0468	0.495000 0.0958286	0.369000 0.0494571	0.676000 0.0607429
0.395000 0.0602	0.769000 0.043	0.503000 0.0905714	0.373000 0.0362857	0.690000 0.0506
0.400000 0.07	0.787000 0.042	0.510000 0.0920286	0.377000 0.0512857	0.704000 0.0539714
0.405000 0.0696	0.806000 0.0394	0.518000 0.0792857	0.382000 0.0422286	0.719000 0.0401857
0.410000 0.0688	0.826000 0.0356	0.526000 0.0805714	0.386000 0.0423429	0.735000 0.0486286
0.415000 0.0744	0.847000 0.0344	0.535000 0.0881143	0.391000 0.0613143	0.752000 0.0487714
0.420000 0.0752	0.870000 0.0328	0.543000 0.0866571	0.395000 0.0598286	0.769000 0.0449143
0.426000 0.074	0.893000 0.0296	0.552000 0.0846857	0.400000 0.0695714	0.787000 0.0438286
0.431000 0.0618	0.917000 0.021	0.562000 0.0832286	0.405000 0.0692571	0.806000 0.0411314
0.437000 0.0848	0.943000 0.0102	0.571000 0.0816	0.410000 0.0686286	0.826000 0.0372371
0.442000 0.087	0.971000 0.0204	0.581000 0.0833143	0.415000 0.0742286	0.847000 0.0360686
0.448000 0.0872	1.000000 0.021	0.592000 0.0771429	0.420000 0.0750286	0.870000 0.0343829
0.455000 0.0872	15.000	0.602000 0.0766286	0.426000 0.0739429	0.893000 0.03106
0.461000 0.0898	0.350000 0.0348286	0.613000 0.0750571	0.431000 0.0618	0.917000 0.02204
0.467000 0.0878	0.353000 0.0449143	0.625000 0.0730286	0.437000 0.0848857	0.943000 0.0105457
0.474000 0.0886	0.357000 0.0394286	0.637000 0.0732857	0.442000 0.0872	0.971000 0.02134
0.481000 0.0898	0.361000 0.037	0.649000 0.0673143	0.448000 0.0876286	1.000000 0.02218
0.488000 0.0818	0.365000 0.0489714	0.662000 0.0688286	0.455000 0.0877714	30.000
0.495000 0.0868	0.369000 0.0533143	0.676000 0.0664857	0.461000 0.0905143	0.350000 0.0291429
0.503000 0.0818	0.373000 0.0391429	0.690000 0.0554571	0.467000 0.0884571	0.353000 0.0374286
0.510000 0.083	0.377000 0.0553714	0.704000 0.0591143	0.474000 0.0894	0.357000 0.0328
0.518000 0.0714	0.382000 0.0455714	0.719000 0.0443714	0.481000 0.0907714	0.361000 0.0306286
0.526000 0.0724	0.386000 0.0457714	0.735000 0.0533429	0.488000 0.0829143	0.365000 0.0404571
0.535000 0.079	0.391000 0.0663143	0.752000 0.0534	0.495000 0.0880571	0.369000 0.0439429
0.543000 0.0776	0.395000 0.0647429	0.769000 0.0492286	0.503000 0.0832571	0.373000 0.0321429
0.552000 0.0758	0.400000 0.0753143	0.787000 0.0480571	0.510000 0.0845429	0.377000 0.0453714
0.562000 0.0742	0.405000 0.075	0.806000 0.0451143	0.518000 0.0728857	0.382000 0.0372857
0.571000 0.0728	0.410000 0.0743714	0.826000 0.0408571	0.526000 0.0740571	0.386000 0.0372571
0.581000 0.0742	0.415000 0.0805143	0.847000 0.0395714	0.535000 0.081	0.391000 0.0538571

0.395000 0.0524571	0.552000 0.0535143	0.917000 0.0104943	0.437000 0.0314	0.637000 0.005524
0.400000 0.0608286	0.562000 0.0523143	0.943000 0.00416029	0.442000 0.0321143	0.649000 0.005524
0.405000 0.0604571	0.571000 0.0510857	0.971000 0.0106114	0.448000 0.0320857	0.662000 0.005524
0.410000 0.0597714	0.581000 0.0519143	1.000000 0.011829	0.455000 0.0319143	0.676000 0.005524
0.415000 0.0645143	0.592000 0.0476286	60.000	0.461000 0.0327143	0.690000 0.005524
0.420000 0.0651143	0.602000 0.0473143	0.350000 0.0212229	0.467000 0.0318857	0.704000 0.005524
0.426000 0.064	0.613000 0.0461143	0.353000 0.0271429	0.474000 0.0320571	0.719000 0.005524
0.431000 0.0534	0.625000 0.0446571	0.357000 0.0236571	0.481000 0.0323714	0.735000 0.005524
0.437000 0.0731714	0.637000 0.0445714	0.361000 0.0220571	0.488000 0.0294571	0.752000 0.005524
0.442000 0.0750286	0.649000 0.0407143	0.365000 0.0289714	0.495000 0.0311429	0.769000 0.005524
0.448000 0.0752	0.662000 0.0414571	0.369000 0.0313143	0.503000 0.0293143	0.787000 0.005524
0.455000 0.0751714	0.676000 0.0398286	0.373000 0.0227714	0.510000 0.0296857	0.806000 0.005524
0.461000 0.0772857	0.690000 0.0327571	0.377000 0.0320571	0.518000 0.0254286	0.826000 0.005524
0.467000 0.0754286	0.704000 0.035	0.382000 0.0262286	0.526000 0.0257714	0.847000 0.005524
0.474000 0.0760286	0.719000 0.0248514	0.386000 0.0261429	0.535000 0.0280857	0.870000 0.005524
0.481000 0.077	0.735000 0.0312771	0.391000 0.0375714	0.543000 0.0275143	0.893000 0.005524
0.488000 0.0702571	0.752000 0.0312714	0.395000 0.0364857	0.552000 0.0267714	0.917000 0.005524
0.495000 0.0743714	0.769000 0.0287571	0.400000 0.0421429	0.562000 0.0261143	0.949000 0.005524
0.503000 0.0701714	0.787000 0.028	0.405000 0.0417143	0.571000 0.0254	0.971000 0.005524
0.510000 0.0710857	0.806000 0.0261686	0.410000 0.0411143	0.581000 0.0258	1.000000 0.005524
0.518000 0.0611714	0.826000 0.0234743	0.415000 0.0442571	0.592000 0.0233143	180.0
0.526000 0.062	0.847000 0.0227371	0.420000 0.0444857	0.602000 0.0233714	0.350000 0.005524
0.535000 0.0676571	0.870000 0.02156	0.426000 0.0436	0.613000 0.0227429	0.353000 0.005524
0.543000 0.0663714	0.893000 0.0193771	0.431000 0.0362286	0.625000 0.0219714	0.357000 0.005524
0.552000 0.0647429	0.917000 0.0131057	0.437000 0.0495143	0.637000 0.0218286	0.361000 0.005524
0.562000 0.0634286	0.943000 0.00567143	0.442000 0.0506	0.649000 0.0196943	0.365000 0.005524
0.571000 0.0620286	0.971000 0.0129657	0.448000 0.0505714	0.662000 0.0202343	0.369000 0.005524
0.581000 0.0631714	1.000000 0.0135714	0.455000 0.0504571	0.676000 0.0194114	0.373000 0.005524
0.592000 0.0582571	50.000	0.461000 0.0517429	0.690000 0.0152314	0.377000 0.005524
0.602000 0.0578857	0.350000 0.023	0.467000 0.0503714	0.704000 0.0167714	0.382000 0.005524
0.613000 0.0565143	0.353000 0.0294286	0.474000 0.0506571	0.719000 0.0100171	0.386000 0.005524
0.625000 0.0548286	0.357000 0.0257143	0.481000 0.0511714	0.735000 0.0147171	0.391000 0.005524
0.637000 0.0549143	0.361000 0.0239429	0.488000 0.0465429	0.752000 0.0150114	0.395000 0.005524
0.649000 0.0503143	0.365000 0.0314571	0.495000 0.0492286	0.769000 0.0136771	0.400000 0.005524
0.662000 0.0513143	0.369000 0.0340286	0.503000 0.0462857	0.787000 0.01326	0.405000 0.005524
0.676000 0.0494286	0.373000 0.0247714	0.510000 0.0468571	0.806000 0.0123543	0.410000 0.005524
0.690000 0.0409429	0.377000 0.0348857	0.518000 0.0401714	0.826000 0.0107771	0.415000 0.005524
0.704000 0.0436857	0.382000 0.0286	0.526000 0.0407143	0.847000 0.0107086	0.420000 0.005524
0.719000 0.0319629	0.386000 0.0285143	0.535000 0.0443714	0.870000 0.0101457	0.426000 0.005524
0.735000 0.0392571	0.391000 0.0409714	0.543000 0.0434	0.893000 0.00898	0.431000 0.005524
0.752000 0.0392571	0.395000 0.0397714	0.552000 0.0422571	0.917000 0.00479143	0.437000 0.005524
0.769000 0.0361171	0.400000 0.0460286	0.562000 0.0412286	0.943000 0.00119543	0.442000 0.005524
0.787000 0.0351829	0.405000 0.0456	0.571000 0.0401714	0.971000 0.00556857	0.448000 0.005524
0.806000 0.033	0.410000 0.0449429	0.581000 0.0406571	1.000000 0.00638571	0.455000 0.005524
0.826000 0.0297771	0.415000 0.0484	0.592000 0.037	180.0	0.461000 0.005524
0.847000 0.0288114	0.420000 0.0486857	0.602000 0.0369143	0.350000 0.005524	0.467000 0.005524
0.870000 0.0273771	0.426000 0.0477429	0.613000 0.0358571	0.353000 0.005524	0.474000 0.005524
0.893000 0.0246886	0.431000 0.0397143	0.625000 0.0346286	0.357000 0.005524	0.481000 0.005524
0.917000 0.0172457	0.437000 0.0542571	0.637000 0.0345429	0.361000 0.005524	0.488000 0.005524
0.943000 0.00796	0.442000 0.0554857	0.649000 0.0312571	0.365000 0.005524	0.495000 0.005524
0.971000 0.0168314	0.448000 0.0554857	0.662000 0.0319143	0.369000 0.005524	0.503000 0.005524
1.000000 0.01752	0.455000 0.0553714	0.676000 0.0305714	0.373000 0.005524	0.510000 0.005524
40.000	0.461000 0.0568	0.690000 0.0245943	0.377000 0.005524	0.518000 0.005524
0.350000 0.0254	0.467000 0.0553143	0.704000 0.0265971	0.382000 0.005524	0.526000 0.005524
0.353000 0.0326286	0.474000 0.0556571	0.719000 0.0174714	0.386000 0.005524	0.535000 0.005524
0.357000 0.0284857	0.481000 0.0562286	0.735000 0.0235514	0.391000 0.005524	0.543000 0.005524
0.361000 0.0265714	0.488000 0.0511714	0.752000 0.0237314	0.395000 0.005524	0.552000 0.005524
0.365000 0.0350286	0.495000 0.0541429	0.769000 0.0216829	0.400000 0.005524	0.562000 0.005524
0.369000 0.0379143	0.503000 0.0509714	0.787000 0.0210143	0.405000 0.005524	0.571000 0.005524
0.373000 0.0276857	0.510000 0.0515143	0.806000 0.0195943	0.410000 0.005524	0.581000 0.005524
0.377000 0.0389714	0.518000 0.0442857	0.826000 0.0173771	0.415000 0.005524	0.592000 0.005524
0.382000 0.0319429	0.526000 0.0448286	0.847000 0.0169257	0.420000 0.005524	0.602000 0.005524
0.386000 0.0318857	0.535000 0.0488	0.870000 0.0159686	0.426000 0.005524	0.613000 0.005524
0.391000 0.0459714	0.543000 0.0478286	0.893000 0.01426	0.431000 0.005524	0.625000 0.005524
0.395000 0.0446857	0.552000 0.0465714	0.917000 0.00884857	0.437000 0.005524	0.637000 0.005524
0.400000 0.0517429	0.562000 0.0454857	0.943000 0.00312257	0.442000 0.005524	0.649000 0.005524
0.405000 0.0513429	0.571000 0.0442571	0.971000 0.0092	0.448000 0.005524	0.662000 0.005524
0.410000 0.0506286	0.581000 0.0449714	1.000000 0.00984	0.455000 0.005524	0.676000 0.005524
0.415000 0.0546	0.592000 0.0410571	90.000	0.461000 0.005524	0.690000 0.005524
0.420000 0.055	0.602000 0.0408571	0.350000 0.0136914	0.467000 0.005524	0.704000 0.005524
0.426000 0.054	0.613000 0.0397429	0.353000 0.01748	0.474000 0.005524	0.719000 0.005524
0.431000 0.0449143	0.625000 0.0384	0.357000 0.01522	0.481000 0.005524	0.735000 0.005524
0.437000 0.0614857	0.637000 0.0383143	0.361000 0.0141543	0.488000 0.005524	0.752000 0.005524
0.442000 0.0629714	0.649000 0.0348	0.365000 0.0185829	0.495000 0.005524	0.769000 0.005524
0.448000 0.0630571	0.662000 0.0354	0.369000 0.0200257	0.503000 0.005524	0.787000 0.005524
0.455000 0.0629429	0.676000 0.0339714	0.373000 0.0145857	0.510000 0.005524	0.806000 0.005524
0.461000 0.0646286	0.690000 0.0276657	0.377000 0.0204857	0.518000 0.005524	0.826000 0.005524
0.467000 0.0630571	0.704000 0.0296857	0.382000 0.0167371	0.526000 0.005524	0.847000 0.005524
0.474000 0.0634571	0.719000 0.0203143	0.386000 0.0166543	0.535000 0.005524	0.870000 0.005524
0.481000 0.0641714	0.735000 0.0264171	0.391000 0.0239429	0.543000 0.005524	0.893000 0.005524
0.488000 0.0584286	0.752000 0.0264743	0.395000 0.0231714	0.552000 0.005524	0.917000 0.005524
0.495000 0.0618571	0.769000 0.02424	0.400000 0.0267714	0.562000 0.005524	0.943000 0.005524
0.503000 0.0583429	0.787000 0.0235771	0.405000 0.0265714	0.571000 0.005524	0.971000 0.005524
0.510000 0.0590286	0.806000 0.0219771	0.410000 0.0261714	0.581000 0.005524	1.000000 0.005524
0.518000 0.0507143	0.826000 0.0196171	0.415000 0.0280857	0.592000 0.005524	
0.526000 0.0514	0.847000 0.0190343	0.420000 0.0282286	0.602000 0.005524	
0.535000 0.0560571	0.870000 0.0180029	0.426000 0.0276286	0.613000 0.005524	
0.543000 0.0548857	0.893000 0.0161286	0.431000 0.0229714	0.625000 0.005524	

Figure 4
Background File

Using these new input files, visible images were then created and compared to the visible images of real tanks. This comparison revealed what at first seemed to be a major problem. The pixels in the Irma images were perfect squares while the pixels in the real imagery were flatter rectangles. An inspection of the specification of the camera used to take the real imagery showed that the pixel size in the images

produced by that camera is 8.4 μm X 9.8 μm . To fix this problem, the Irma simulation had to be configured to match the real camera specifications. The Irma Sensor Configuration File, which describes the sensor used in the simulation, defines the pixel size (detector size) in radians rather than microns. Therefore, the ratio from the camera specifications was used to change these values so that the image size and shape would match. Figure 5 is the original Sensor Configuration file and Figure 2 is the improved one, with the altered detector sizes highlighted.

1	sensor mode 1-ir 2-tex 3-ir+tex 4-rng 5-ir+rng 6-tex+rng 7=1+2+3
128 128	number of detectors in n1 and n2 directions
.000436 .000436	detector size in n1 and n2 directions (rad)
0.0 0.0	size of gap between detectors in n1 and n2 directions (rad)
1 1	image oversampling factors in n1 and n2 directions
0.00045 11	rms width spatial resolution (radians) & no. pix in mask
0.0 0.0 0	sensor noise parameters (a, b, seed)
0.0 0.0 0 0	sensor scan parameters (scan rate, cutoff freq, masksiz)

Figure 5
Sensor Configuration File

1	sensor mode 1-ir 2-tex 3-ir+tex 4-rng 5-ir+rng 6-tex+rng 7=1+2+3
128 128	number of detectors in n1 and n2 directions
.000436 .000373714	detector size in n1 and n2 directions (rad)
0.0 0.0	size of gap between detectors in n1 and n2 directions (rad)
1 1	image oversampling factors in n1 and n2 directions
0.00045 11	rms width spatial resolution (radians) & no. pix in mask
0.0 0.0 0	sensor noise parameters (a, b, seed)
0.0 0.0 0 0	sensor scan parameters (scan rate, cutoff freq, masksiz)

Figure 6
New Sensor Configuration File

Now the Irma simulation was correctly configured to produce visible images that were comparable to existing real images taken by a video camera. Figure 7 and Figure 8 are Irma created visible images of a M48 and M60 respectively. Obviously, the images do not look very similar in color and overall appearance. During the optical filter generation, this proved to be a major drawback to these synthetic images. Since the Irma software is not really designed to produce real images, "tricking" it to do so does not produce very consistent results. This problem was taken care of as edge enhancement and binarization techniques were used during the filter generation process.

The only other alteration to the Irma input files was another change to the Sensor Configuration File. This change, however did not happen until later in the experimentation, during the optical filter generation. To smooth the images it creates, Irma's Sensor Configuration File specifies a pixel



Figure 7
Visible Irma M48

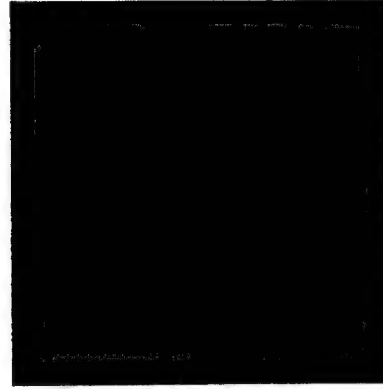


Figure 8
Visible Irma M60

convolution mask that is applied to the image. Originally, the Sensor Configuration File used an 11 pixel mask. When applied to the images created for these experiments, that mask causes an outline of very dark, yet not black, pixels to surround the target due to the black pixels of the background being included in the convolution. This becomes very obvious when the target is placed on a lighter background, as in Figure 9. At first, this seemed to pose a major problem as the images are hard to size and the filtering software uses all the pixels, even though they are hard to visually recognize. Different convolution masks, using 1 and 3 pixels, were tested. Without the smoothing effect, the images created are very blocky, which really does not work well in comparison to the real images. The images created with the 11 pixel mask turned out to be the most useful as edge enhancement and binarization techniques were applied, which will be discussed in greater detail later.

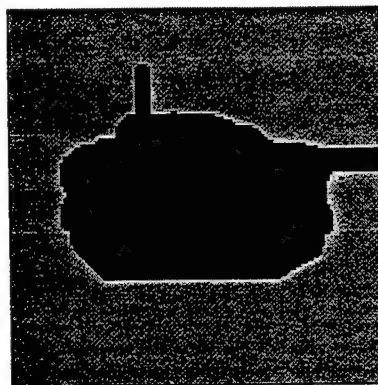


Figure 9
Irma M60
Background Intensity=128.0

It is significantly important that the targets be close to the same size for the optical correlation. In developing this new technique of image generation, a desirable factor is that the simulation image acquisition be a good recreation of the real image acquisition. Unfortunately, Irma does not have a set scale of measurements for its simulation and is dependent upon the facet model description. Therefore the only way to match the image sizes is by the actual size of the image in pixels. Once the width in pixels of the real image is found, the Irma Eyepoint File, through the positioning of the seeker, can be adjusted to match this size. Through this method, the synthetic images created by Irma can be matched very closely to any set of real imagery.

II. Optical Filter Design and Testing

Once a method of image synthesis utilizing Irma was developed, the next step in the experimentation was to create optical filters using the Irma images. The filters were created on a PC with a bank of 30 transputers using the software tools developed by Dr. Robert R. Kallman for spatial filter design. All of the filters created are fifteen state spatial filters using training set of twenty-one images.

A complete set of optical filters for a munition guidance system would be capable of covering a range from 20° to 50° elevation in a complete 360° rotation. Figure 10 and Figure 11 shows the image range of a set of optical filters for a single target. One filter, however, cannot encompass every angle and aspect in this range. Besides software and hardware limitations of the filter design tools, a filter becomes less accurate as the number of images used to make it increases. Therefore, the ranges needed would be covered by several filters rather than just one. For these experiments, an image set of twenty-one images was created for each filter, ranging from 80° to 100° in rotation. Figure 12 shows this range.

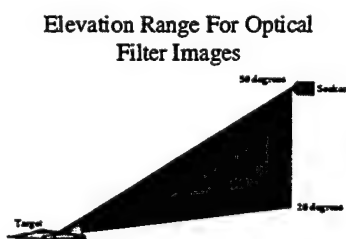


Figure 10

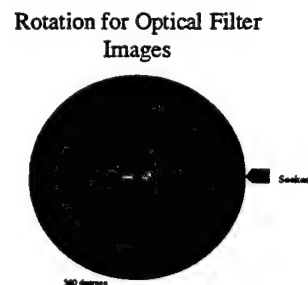


Figure 11

Elevation and Rotation of Test Sets

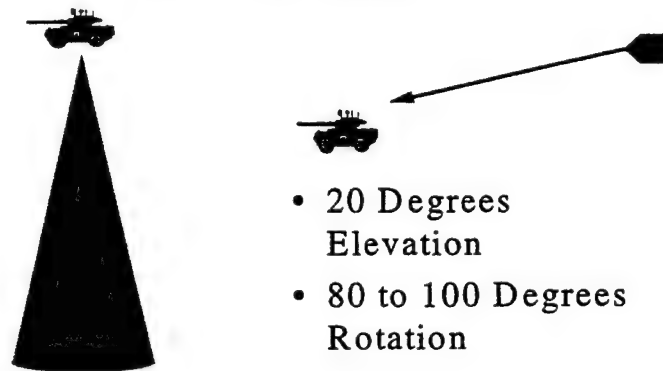


Figure 12

All of the experimental optical filters built for during this research endeavor used the Irma M60 images. M60 and M48 were chosen for testing the Irma images for filter use because both real and scale model imagery already existed of these targets. The M60 was chosen above the M48 because its facet model seemed to match the real imagery closer than the M48. Initial tests using the Irma M48 imagery to create optical filters showed that the M48 facet model may not be compatible to the real imagery. This proved to be a limiting factor of using Irma to generate imagery. The facet models used to create the synthetic imagery needs to be very close to the real thing, else it makes it difficult to correlate.

The first filter built using the Irma imagery was based on the visible M60 images, placed on a background of intensity 128.0, as in Figure 9. It was built using the zero mean intensity phase-encoded input method. When tested against the real M60 imagery, it yielded small correlation peaks, which is good. Unfortunately it also yielded small peaks when tested against the real M48 imagery, sometimes actually matching the M48 better. Figure 13 and Figure 14 are the real tank images. Figure 15 and Figure 16 are the correlation peaks at 90°. These results, coupled with the failure of the initial M48 filters showed that some type of image processing would have to be applied to the Irma images for this experiment to be successful. Using the input files described in this report, Irma does not produce very consistent visible imagery, as evidenced by Figure 7 and Figure 8. The differences in intensity and color of the Irma

imagery and real imagery posed a problem using this correlation method. Therefore, edge enhancement and binarization techniques were investigated as possible solutions.

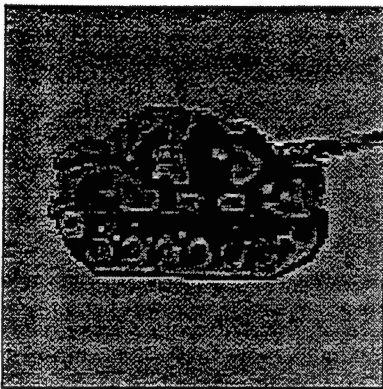


Figure 13
Real M60
Background Intensity=128.0

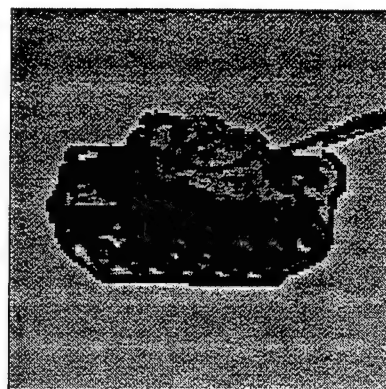


Figure 14
Real M48
Background Intensity=128.0

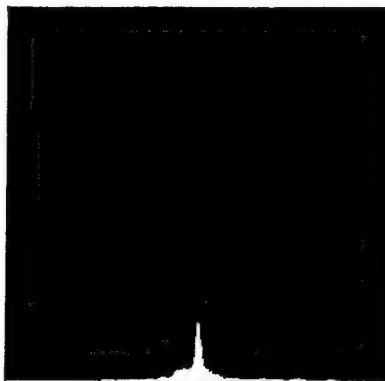


Figure 15
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M60

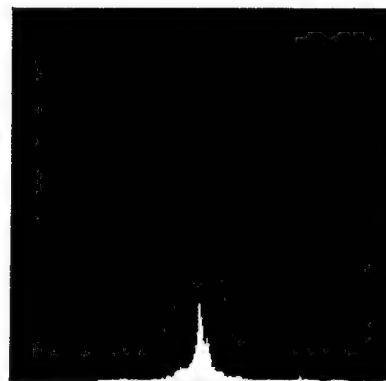


Figure 16
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M48

It was hypothesized that better filters could be built using edge enhanced, binarized imagery because only an outline of the image would be used in the correlation, thereby eliminating the effects of image color and intensity. After experimenting with many different methods of edge enhancement and binarization, one method was finally settled on that yielded very good outlines.. It was found that if the image is place on a white background before the processing, the outline comes up much better, with no extraneous garbage on the inside. The outline of very dark pixels surrounding the Irma visible images after the convolution smoothing discussed earlier in the report turned out to be an advantage here as it helps bring forth the outlines real well. Sobel edge enhancement is used, along with global binarization of the edge

enhanced image using the maximum scaling mode with a scaling factor of 0.5. Figure 17 presents this method.

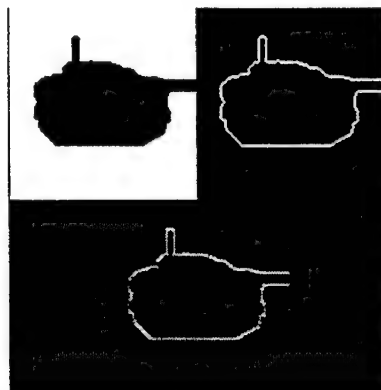


Figure 17
Edge Enhance, Binarized Irma M60

The final images, like the tank outline in the top right corner of Figure 17, were made into a filter. It was created using the standard amplitude input method. The same process of edge enhancement and binarization was performed on the real imagery and the filters were tested. Again, the correlations from the M60 and M48 were very close as the filter has a hard time distinguishing between the two tanks. Figure 22 is a chart plotting the maximum detector correlation intensity/threshold (the height of the correlation peak from 0 to 1, the indication of how well the images match) at the different angles of the target set. It makes it very clear that there is not enough difference in correlation for a detector to distinguish between the tanks. Therefore, the real M48 image set was added into a new filter as a false target to train the filter to discriminate against M48 images. As a result the correlation intensity of the tests against the real M60 dropped slightly, but there is no correlation at all against the M48. Figure 25, the same type of chart as Figure 18²², shows an obvious distinction between the M60 and M48 correlations.

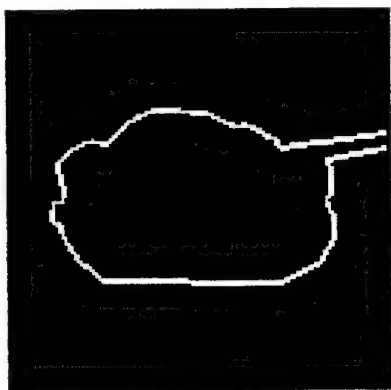


Figure 18
Edge Enhanced, Binarized Real M60

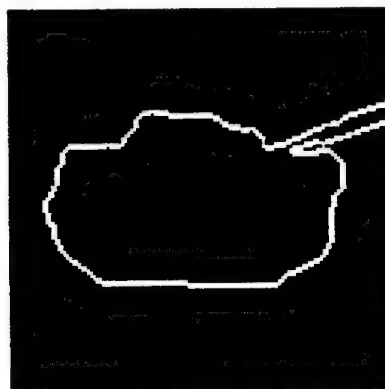


Figure 19
Edge Enhanced, Binarized Real M48

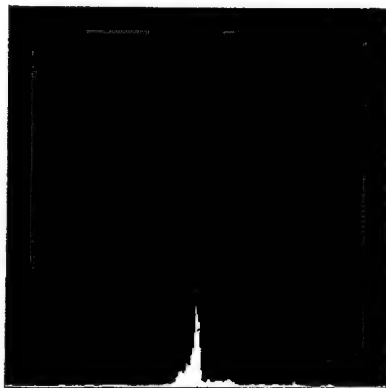


Figure 20
Irma M60 Filter vs. Real M60

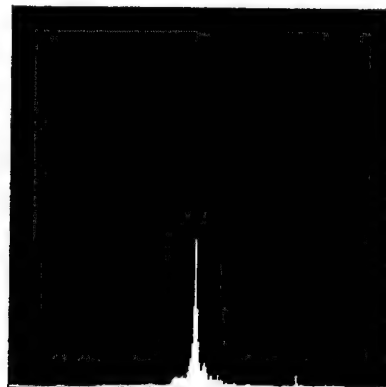


Figure 21
Irma M60 Filter vs. Real M48

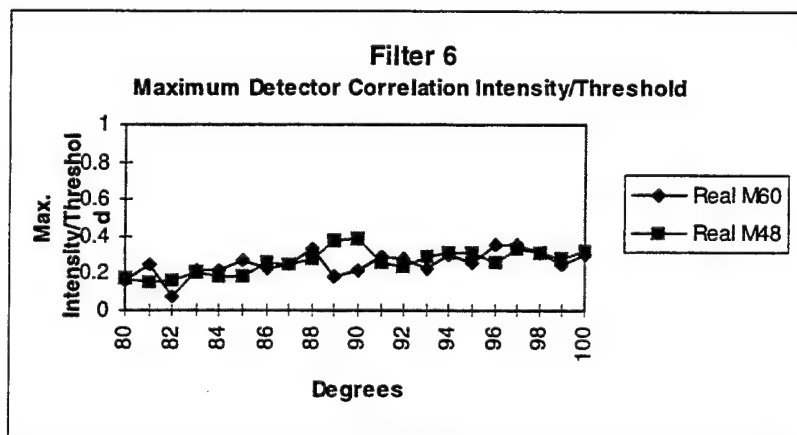


Figure 22



Figure 23
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M60
False Target: Real M48

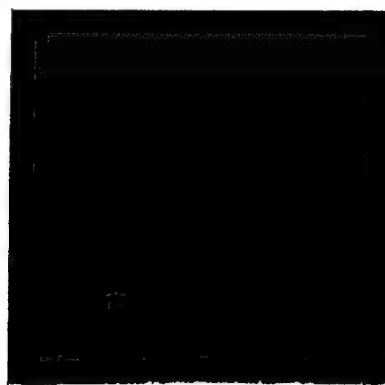


Figure 24
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M48
False Target: Real M48

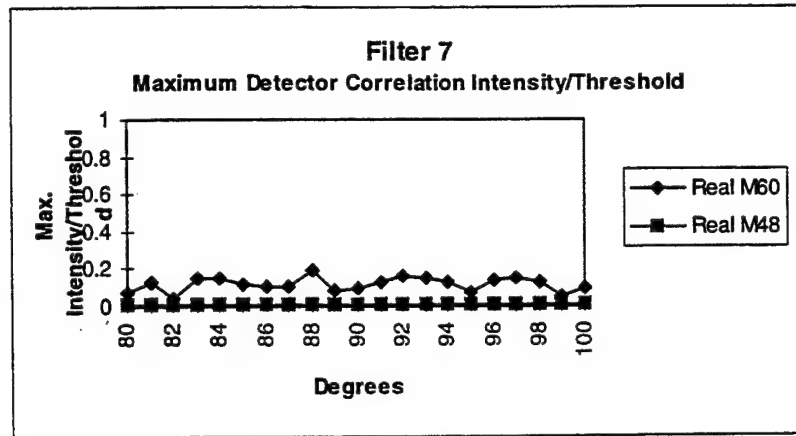


Figure 25

During the development of this filter and its testing, it was discovered that not edge enhancing the image before the binarization results in simply a black shape of the target. Theorizing that this type of image might be well suited for the filter design, this method was more fully explored. Figure 26 shows the results of this experimentation. To make these images, the target is once again placed on a white background. Then it is binarized using global binarization with the average scaling mode using a scaling factor of 1.0.

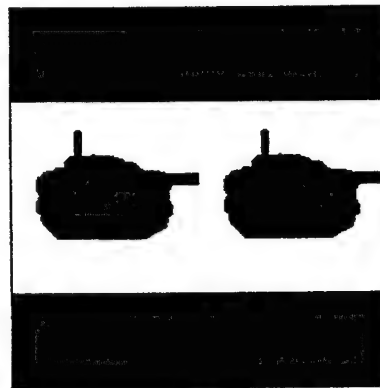


Figure 26
Binarized Irma M60

The same technique was applied to the real imagery. Again, a filter was created from the Irma imagery and tested against the real imagery. The result was a correlation peak higher than any obtained to that point. Unfortunately, the M48 correlated very highly again. Therefore, real M48 imagery was again used as a false target. The intensities were again lowered when the filter was tested against the M60, but the M48 did not correlate at all.

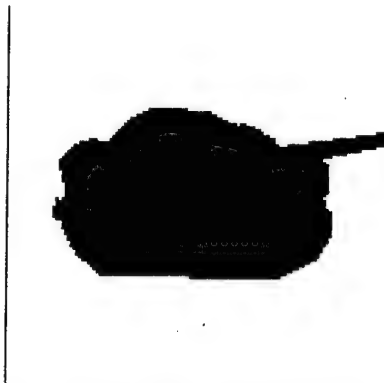


Figure 27
Binarized Real M60



Figure 28
Binarized Real M48

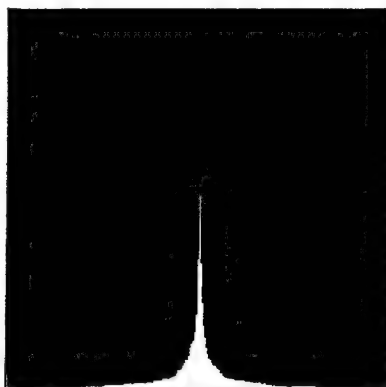


Figure 29
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M60

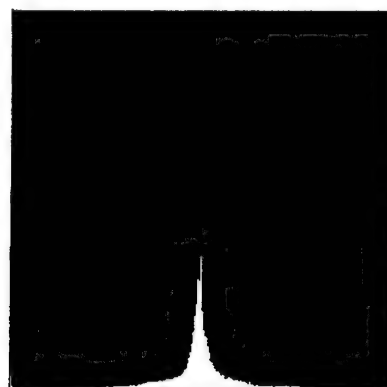


Figure 30
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M48

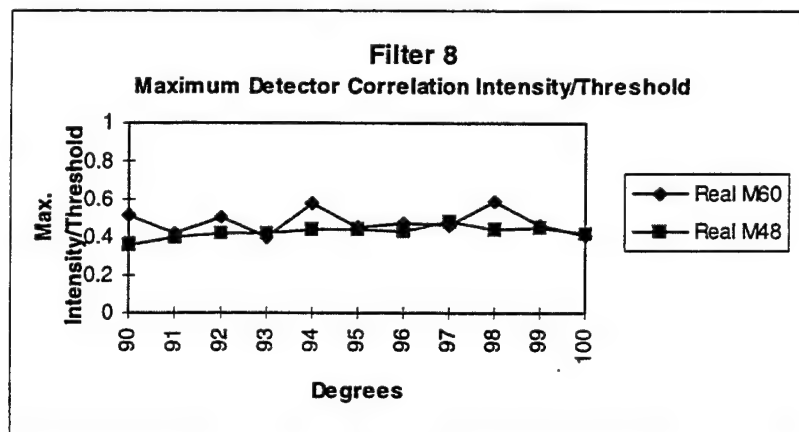


Figure 31¹

¹ This filter was created using only a 90° to 100° rotation

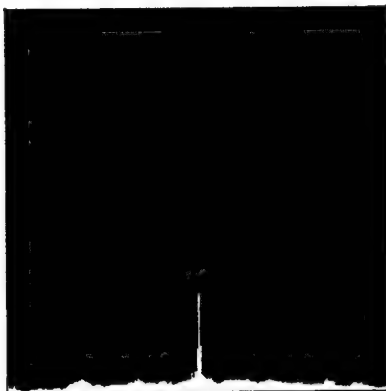


Figure 32
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M60
False Target: Real M48



Figure 33
Correlation Plane at 90°, South to North View
Irma M60 Filter vs. Real M48
False Target: Real M48

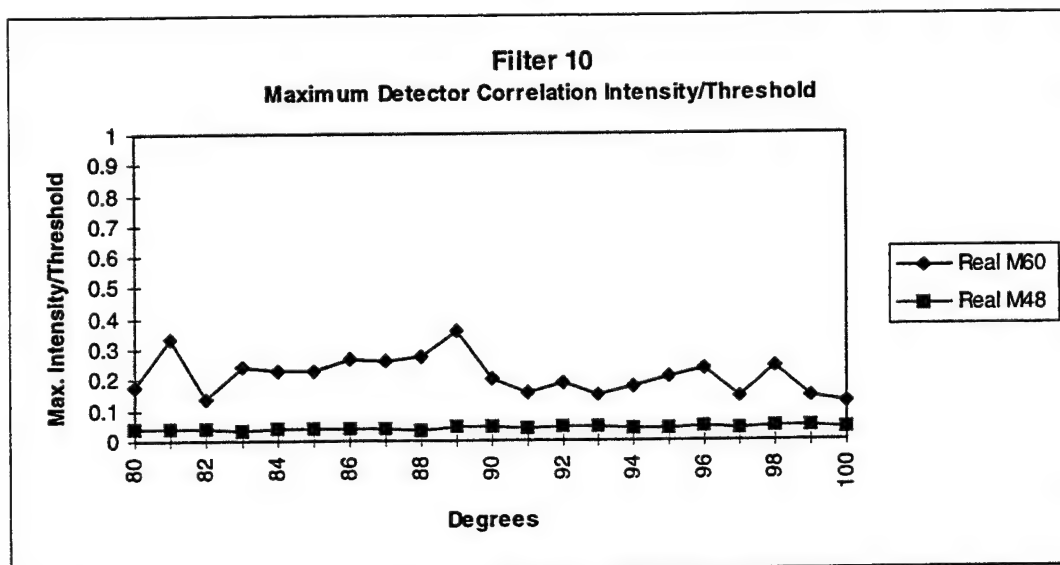


Figure 34

As Figure 34 shows, there is an obvious degree of difference between the correlation of the M60 and M48, enough for a detector to distinguish between them.

In conclusion this experiment has shown that filters built with computer generated images are comparable to those generated by scale models. It has also shown that care must be taken to generate the images to closely match the real target. The capability to include false target information in the filter illustrates the requirement that the generated filters be very robust to distinguish within a class of targets. Further research will examine the filter response in the optical processing hardware.

Aerodynamics of a Generic Miniaturized Munition Technology (MMT) Airframe

Elizabeth Anne Walker

**Niceville Senior High School
800 E. John Sims Parkway
Niceville, FL 32578**

**Final Report for:
High School Apprentice Program
Wright Laboratory**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC**

and

Wright Laboratory

August 1995

AERODYNAMICS OF A GENERIC MINIATURIZED MUNITION TECHNOLOGY (MMT) AIRFRAME

Elizabeth Anne Walker
Niceville Senior High School

Abstract

AP95 and PRODAS, two computer programs, are aeroprediction codes used to predict the aerodynamic coefficients of certain projectiles. The projectile studied for the Miniaturized Munition Technology (MMT) Program has three tail fins, two canards, and a two segment nose. Neither program alone could handle the advanced aerodynamic configurations. AP95 has only the option of two or four tail fins, and PRODAS can not operate with canards on nose segments. Results were then obtained from AP95 for four fins with and without canards and from PRODAS for three and four fins (no canards). The results were then combined by converting AP95's results from three fins to four using the results of PRODAS. The aerodynamic coefficients were found for 0, 5, and 10 degrees angle of attack for mach number ranging from 0.1 to 5. AP95 and PRODAS were compared and advantages and disadvantages to each program were found. A method of prediction for complex configurations was also developed.

AERODYNAMICS OF A GENERIC MINIATURIZED MUNITION TECHNOLOGY (MMT) AIRFRAME

Elizabeth Anne Walker

Over the centuries, aerodynamics has become a subject of intellectual beauty, developed by such great minds as Newton, Euler, and the Wright brothers. From reducing the drag of a sailboat centuries ago to the designing of the first plane, aerodynamics has played a significant role in the past and is vital to the ultimate success of many designs in today's world. Before 1971, there were three ways to obtain aerodynamics of a given configuration -- perform flight tests of a full-scale configuration; design, build, and test a wind tunnel model over the flight range of interest; or empirically estimate the aerodynamics by utilizing existing handbooks, wind tunnel data reports, and theoretical analysis. These processes are time consuming and often costly. The Navy began to develop an alternative by 1971 in which a computer program could be used to compute the aerodynamics. The code was based on appropriate approximations to the Euler and Boundary Layer Equations. It has been updated six times in an add-on fashion since then in an attempt to meet the requirements of the tactical weapons community.

AP95 is an aeroprediction code that is used to determine the aerodynamic coefficients of certain projectiles. The Naval Surface Warfare Center developed this engineering code, and AP95 is the latest version. It has been updated from the 1993 version and extended to angles of attack (AOA) greater than 30 degrees and is now more compatible to a personal computer (PC). Comparisons of the code's predictions to actual experimentation have been performed, and AP95's results in general had an average accuracy level of ten percent for axial and normal force coefficients and a four percent level of accuracy for body length for center of pressure, with some exceptions due to the fact that nonlinearities caused by interval shock interactions were not accounted for. AP95 is a semiempirical code in which its predictions come from a blend of approximate theories and empirical estimates.

Configurations on AP95 are classified into four categories: Body-Alone, Wing-Alone, Body-Tail, or Canard / Wing - Body - Tail. Data is then inputted into three sections -- geometry, free-stream conditions, and options. The geometry input screen is further split by component parts -- nose, afterbody, boattail / flare, tail, canard / wing, body roughness, and general data. The free-stream conditions involve changes in angle of attack (alpha sweep), surface deflection angle (delta sweep), altitude (altitude sweep), and mach numbers (mach sweep). Inputs then

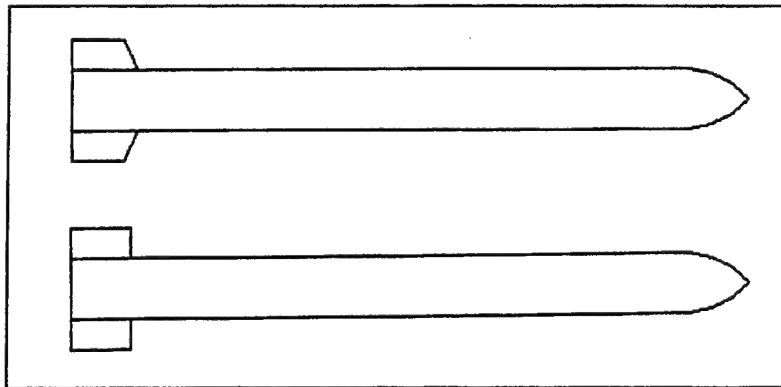


Figure 1. Clipped Delta and Square Fins

include, for example, initial angle of attack, final angle of attack, and interval size. Two Body-Tail Configurations (Fig. 1) were entered first. Results were obtained and then compared to some tests that had been run by the Navy. This was done in order to verify that program was

running properly. The results came out the same after a number of revisions.

PRODAS, PROjectile Design and Analysis System, is another aeroprediction code and has been used by the defense community for over twenty years. Projectiles are entered by elements, in which each element is defined by left diameter, right diameter, length, reference length, density (which can be entered by material), element code, and radius. The element code refers to the specific component of the projectile, such as the body, ogive (nose), boattail, fin, or flare. After creating a projectile, PRODAS can calculate the physical properties and analyze the aerodynamics, stability, and trajectory. Six cases (Fig. 2) were first entered. The first two were the clipped delta

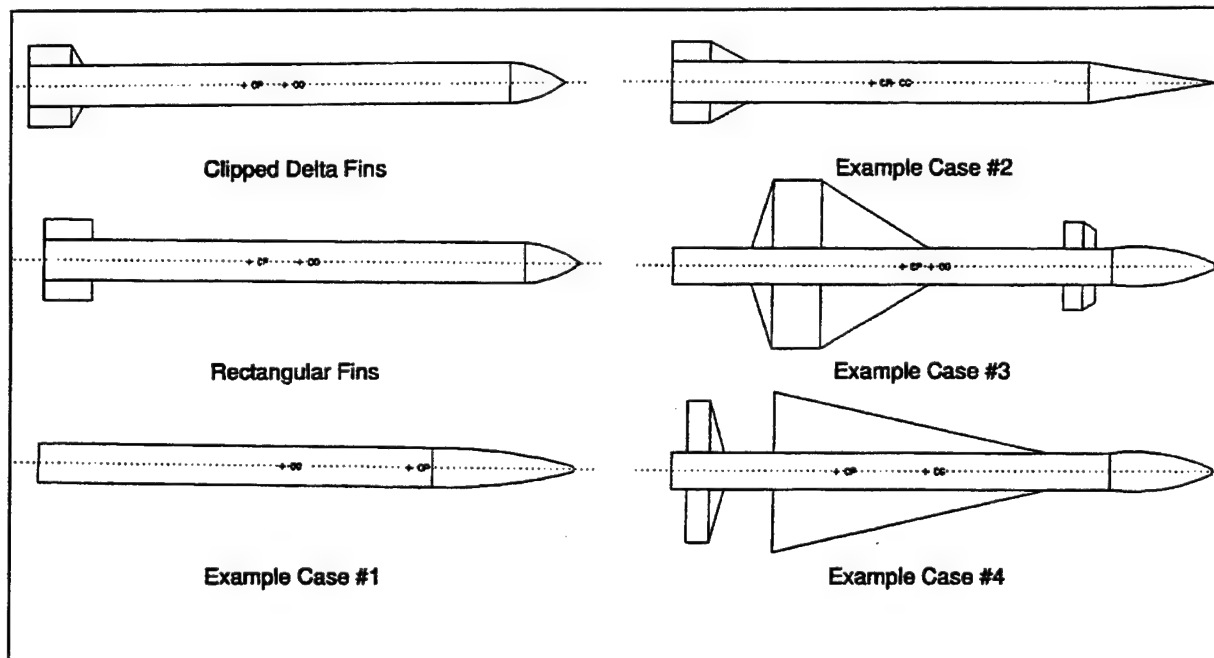
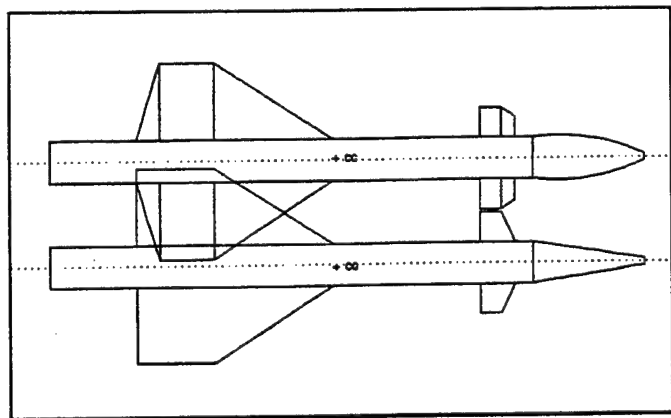


Figure 2. Six Cases on PRODAS

and square finned configurations and the other four were example cases from AP95. Some of the results of AP95 and PRODAS were compared (in Excel). On PRODAS, the aerodynamic coefficients come from a large database, which comes from previously tested designs. During the Aero/Stability Analysis, the program tries to match the



inputted body components to those in its database. If the design is rather complex or nonconventional and does not match, then the results might not be reliable.

Figure 3. Match of Example Case #3 from PRODAS Database

Both programs are written in FORTRAN and are used to evaluate design performance. Results from AP95 and PRODAS, drag (CD), normal force coefficient (CN), and pitching moment (CM) for 0.01, 5.01, and 10 degrees angle of attack, were compared in Excel. AP95's results might not be reliable over ten degrees angle of attack. Results from PRODAS may not be reliable for cases that have large wings. These unreliabilities showed up in comparisons of the example cases.

The Miniaturized Munition Technology (MMT) Program's purpose is to develop a small smart bomb. Accuracy is improved by creating a smaller warhead. Multiple bombs can be launched, with each one having an independent target. The bombs are guided by the Global Positioning System (GPS). The airframe has advanced aerodynamic concepts, in which neither program alone, AP95 or PRODAS, could handle the configurations of the

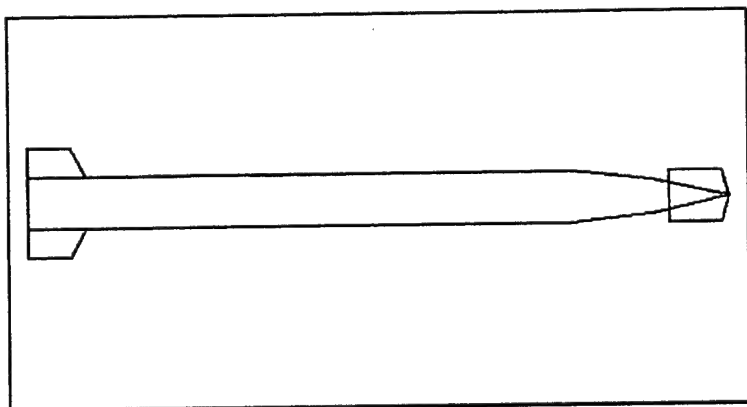


Figure 4. MMT Airframe on AP95

airframe studied. The projectile has a two segment nose, two canards on the nose, and three tail fins. The projectile was first entered into AP95 (Fig. 4). The program could handle the two canards on the nose and the two segment nose, but it only has the option of two or four tail fins.

The projectile was then entered into PRODAS (Fig. 5 & 6). PRODAS could handle the three tail fins, but with a canard on the nose, a program error occurred during the aero/stability analysis. Results were then obtained from

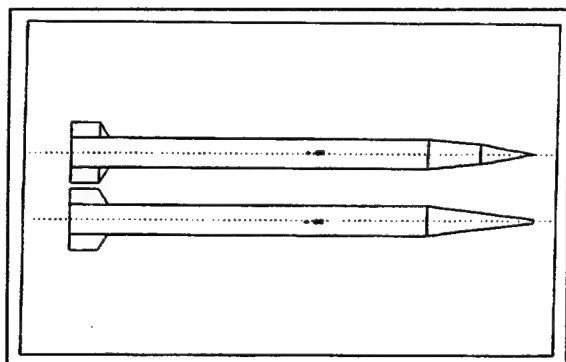


Figure 5. PRODAS Match of MMT Airframe

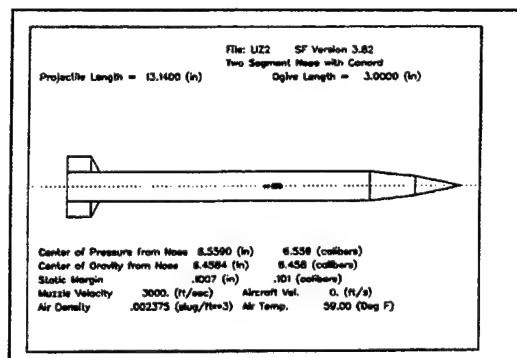


Figure 6. Summary of PRODAS Results

AP95 for four fins with and without canards and from PRODAS for three and four fins (no canards). Comparisons of the configuration with four fins and no canards were then made. Chart 1 is one of these comparisons. The results of AP95 and PRODAS were similar, except at ten degrees angle of attack there were some differences probably due to AP95's unreliabilities at or above ten degrees angle of attack.

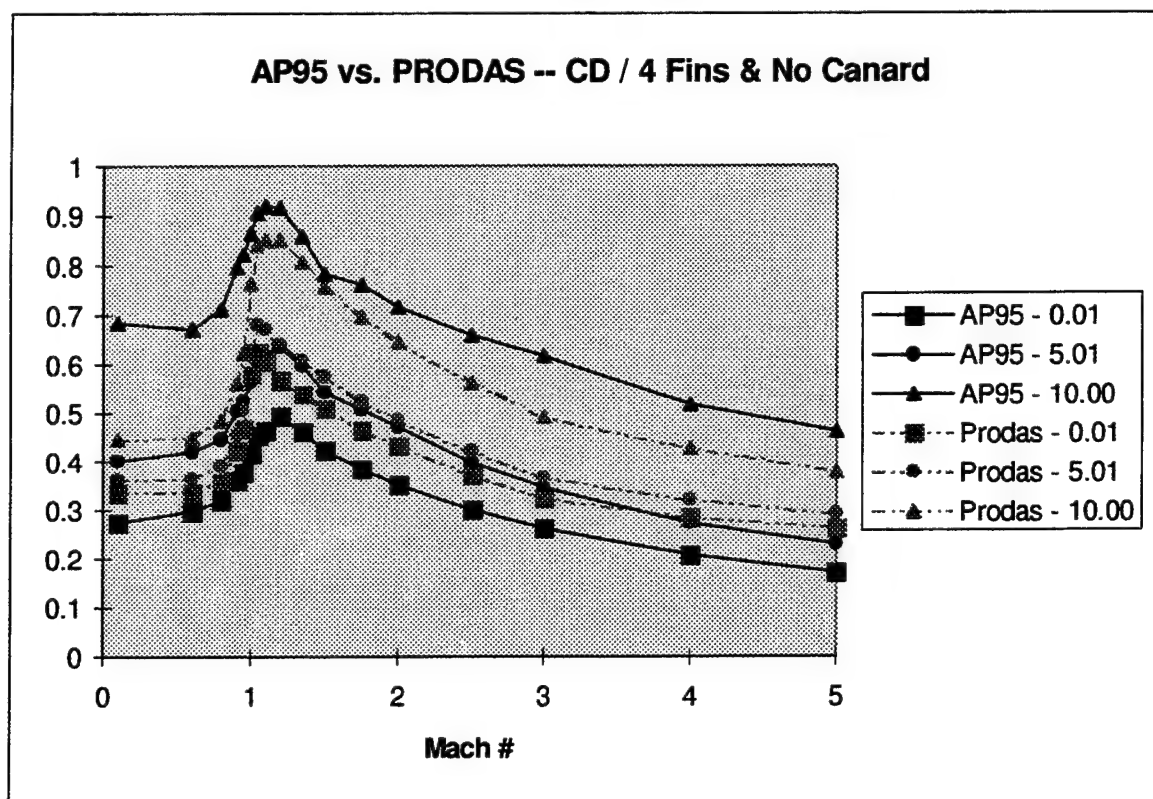
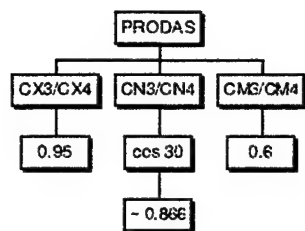


Chart 1. Comparison of Drag (CD) for 4 Fins and No Canard - AP95 vs. PRODAS



**Table 1. Constants from
PRODAS**

The results then had to be combined. PRODAS's results were entered into a spreadsheet, and constants were calculated. AP95's data was then entered into the spreadsheet, and the constants were applied to the appropriate factors. The aerodynamic coefficients were determined for three fins. Table 2 shows the final coefficients for drag (CD), lift (CN), and pitching moment (CM), for 0.01, 5.01, and 10 degrees angle of attack at mach numbers 0.1 to 5.

In conclusion, the aerodynamic coefficients of the MMT Airframe were found. AP95 and PRODAS were compared and advantages and disadvantages to each program were found. A method of prediction for complex configurations was also developed.

Mach #	0.01 Degrees			5.01 Degrees			10 Degrees		
	CD	CN	CM	CD	CN	CM	CD	CN	CM
0.1	0.301225	0.00175	0.0001	0.365925	0.954523	0.33706	0.4275	2.22014	0.81664
0.6	0.301125	0.00148	0.00006	0.3662	0.851776	0.25364	0.42475	1.863013	0.51896
0.8	0.323725	0.00158	-0.00018	0.3868	0.883509	0.16586	0.4429	1.985451	0.32952
0.9	0.36715	0.00167	-0.00038	0.44175	0.943714	0.08158	0.50775	2.104929	0.1658
1	0.422075	0.00175	-0.00062	0.5	0.974058	0.0202	0.5687	2.17062	0.105
1.05	0.4599	0.00184	-0.00054	0.5404	1.005667	0.04724	0.610625	2.195963	0.1376
1.1	0.471075	0.00184	-0.00052	0.5518	0.992598	0.05994	0.622625	2.190608	0.15638
1.2	0.4971	0.00203	-0.0009	0.569325	1.028348	-0.0386	0.61615	2.233163	0.03908
1.35	0.46705	0.00211	-0.0011	0.5309	1.015629	-0.0835	0.57155	2.098073	0.02154
1.5	0.42725	0.00203	-0.00096	0.479	0.946043	-0.04902	0.51035	1.982826	-0.00926
1.75	0.38885	0.00175	-0.00072	0.4493	0.889931	-0.01908	0.49015	1.939907	0.03086
2	0.356975	0.00149	0.00002	0.42415	0.753197	0.31518	0.473	1.79344	0.74784
2.5	0.333075	0.00151	0.00006	0.34765	0.743629	0.31044	0.383625	1.922854	0.79478
3	0.264225	0.00142	0.00012	0.296	0.721096	0.2828	0.323075	1.992974	0.73046
4	0.20775	0.00106	0.002	0.221925	0.695486	0.25932	0.233325	1.880804	0.70756
5	0.1721	0.00294	0.0048	0.179	0.687049	0.39946	0.19255	1.783412	0.85702

Table 2. Final Aerodynamic Coefficients for a Generic MMT Airframe

Bricks, Moments, and a Tetrahedral Search

Josh A. Weaver

Niceville Senior High School
800 E. John Sims Parkway
Niceville, FL 32578

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

Bricks, Moments, and a Tetrahedral Search

Josh Weaver
Niceville High School

Abstract

My first project of this summer, Bricks, was to study the introduction of fragments into the continuous media modeled by a Finite Element Mesh. This was done by identifying the exterior nodes of a given subvolume of the mesh. In addition to writing the actual program, I designed a plotting utility that was used to visually check the accuracy of my algorithm.

Another summer project was to revise and document Moments, my main project from last summer. It consists of a Graphical User Interface for a FORTRAN program, written in the 1980s, that calculated the weight, center of gravity locations, and polar and transverse mass moments of inertia for axisymmetric projectiles.

My main project of this summer was to develop a translator program to convert data files from the EPIC hydrocode into the proper format for SpyGlass Slicer for the PC. The conversion was from an unstructured mesh in EPIC to the structured format required by Slicer. The project was complicated by speed and file size problems.

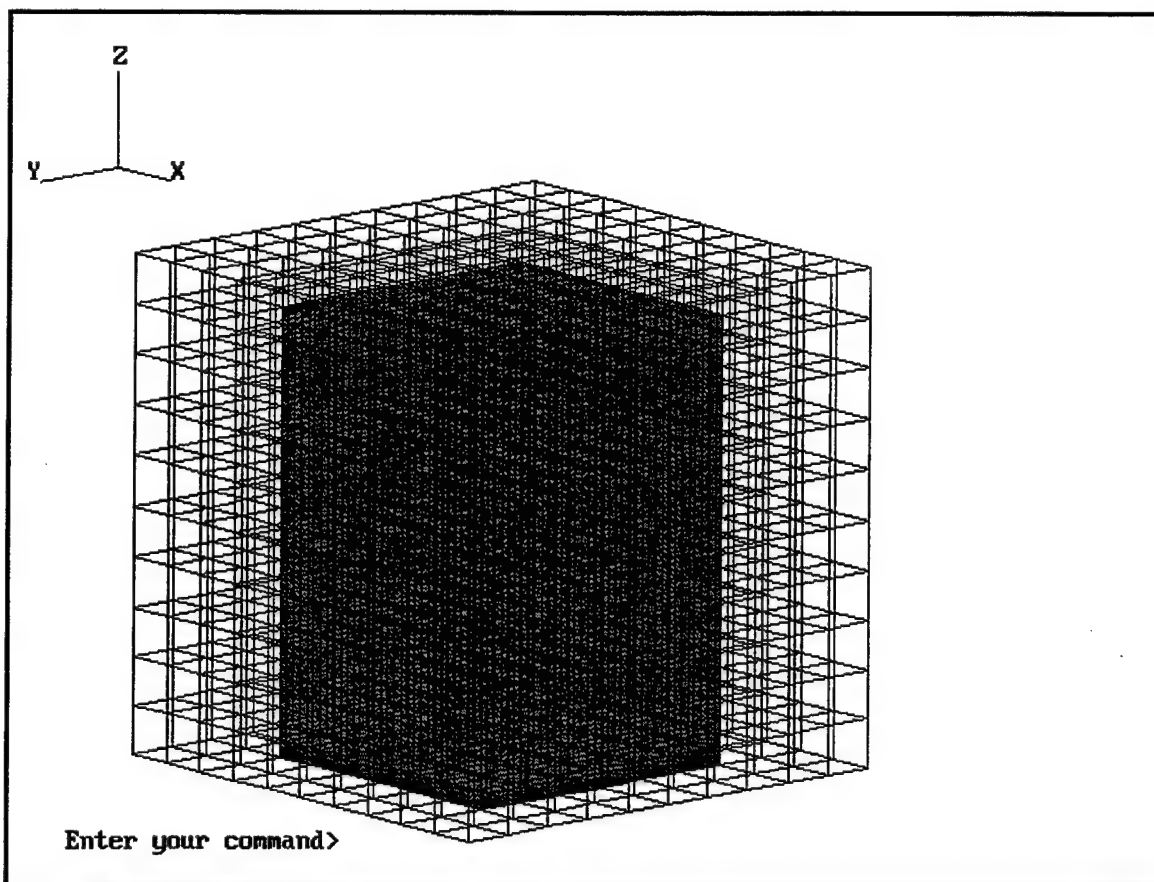
Additionally, I helped design the homepages for the Warheads branch, wrote two data conversion programs, and cataloged reel-to-reel and VHS videos.

This was my second summer working for Mr. Michael E. Nixon in the Computational Mechanics Section in Wright Laboratories located at Eglin, AFB Florida. Over the course of the summer I worked on three major projects --Bricks, Moments, and a Tetrahedral Search -- along with other tasks assigned to help members of the Computational Mechanics Section.

The first project that was assigned to me this summer was to design a program that would be used to test the introduction of fragments into the continuous media modeled by a Finite Element Mesh. The program would design a three dimensional mesh composed of bricks and then identify a given subvolume inside that mesh. Most of the program development centered around creating the best algorithm to accurately 'fit' a cubic volume around a point inside the mesh. The problem was complicated by the nature of the mesh, which was cubic and had a non-linear spacing scheme.

The algorithm that I finally settled on started at the center of the target volume and expanded along each axis, testing for the proper volume as it went. If the best cubic dimensions found were not within the allowed accuracy range, a second algorithm was applied that modified the previous cubic results and approximated the volume into a rectangular solid.

In order to test my results, I created a program that would model the data file and display the target volume inside the original mesh. In the end, I wound up with a utility program that would allow the user to view any mesh defined of bricks, rotate it, and choose between different molding methods to render the data. An example of PcPlot is shown in Figure 53-1.

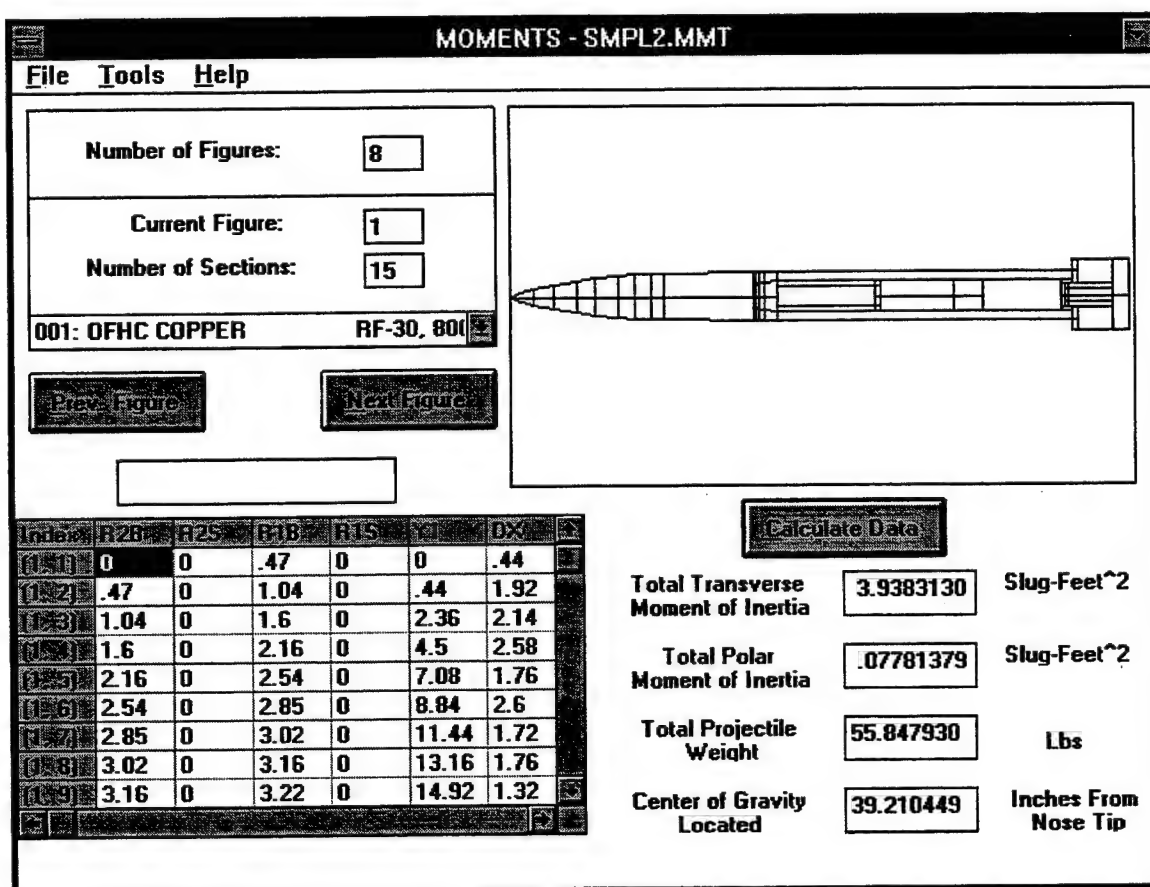


(Figure 53-1) An example of a PcPlot display. In this case, you can tell that the second algorithm was used because of the non-cubic form of the selected volume.

A second task was to document and fine-tune Moments, my project from last summer. Moments is a utility program designed to calculate weight, center of gravity location and the polar and transverse mass moments of inertia for axisymmetric projectiles. My program was based on a FORTRAN utility written in the summer of 1982, which lacked many of the conveniences of more modern 'windowed' programs.

Last summer, I designed a graphical user interface (GUI) for the Moments program to make it easier to use on modern computers. However, since last summer, I have learned many new techniques in computer programming, and I realized that some of them could be used to improve Moments. Also, over the course of the year, users of Moments had found a few bugs that needed to be corrected. Rather than just fine-tuning Moments, I decided to completely rewrite the program using the new programming methods, fix the bugs that were found, and generally improve upon the whole program.

The results are quite promising. The program logic became much more concise using more advanced structures, and the features that I added gave the user much more flexibility in their manipulation of data. Some of the capabilities I added include: on-line help, better error-checking and warning messages, additional tools, and the ability to export data into other programs. Additionally, I provided full documentation for the program, something that had been requested during the year.



Index	R2B	R2S	R1B	R1S	Y	DX
1	0	0	.47	0	0	.44
2	.47	0	1.04	0	.44	1.92
3	1.04	0	1.6	0	2.36	2.14
4	1.6	0	2.16	0	4.5	2.58
5	2.16	0	2.54	0	7.08	1.76
6	2.54	0	2.85	0	8.84	2.6
7	2.85	0	3.02	0	11.44	1.72
8	3.02	0	3.16	0	13.16	1.76
9	3.16	0	3.22	0	14.92	1.32

Calculate Data

Total Transverse Moment of Inertia: 3.9383130 Slug-Feet²

Total Polar Moment of Inertia: .07781379 Slug-Feet²

Total Projectile Weight: 55.847930 Lbs

Center of Gravity Located: 39.210449 Inches From Nose Tip

(Figure 53-2) Picture of main screen in Moments-95. The GUI combines the data input, calculation, and display into a single, easy to use interface.

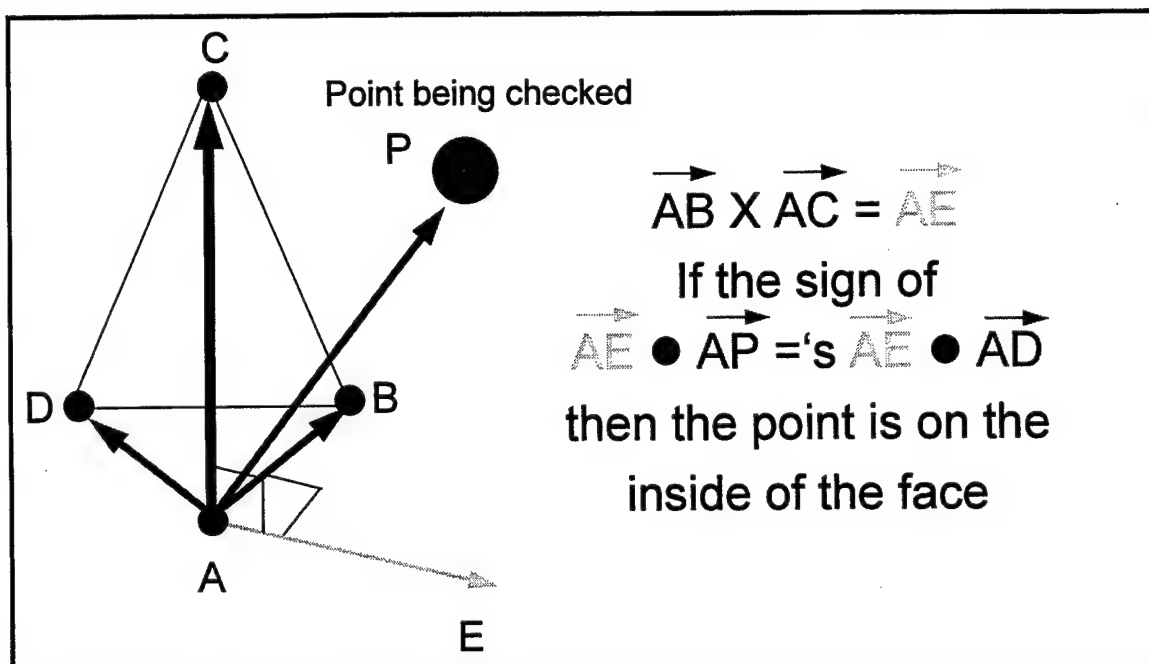
My major project of this summer was to take a PATRAN neutral file from the EPIC hydrocode and manipulate it so that it can be easily viewed on the PC using SpyGlass Slicer. Slicer requires that its data be represented in a ordered cubic form with constant spacing, while EPIC's data is stored as an unstructured mesh. My program reads in EPIC's data file and restructures the data so that Slicer can analyze it. When plotted in Slicer, the data file produced by my program shows the user a 3D image of the

original EPIC calculations. The user can then use the tools available in Slicer to rotate the image, display a slice of the data file, or even display only particular ranges of data as iso-surfaces.

In order to write this program, I first had to learn how the data was stored. EPIC represents objects as a series of tetrahedral elements, each of which contains the address of its nodes, a material type, and six element data values.

The beginning program logic was simple - I had the program create a cubic mesh that surrounded the EPIC object(s) and then checked each node of the mesh to see if it was inside a tetrahedron. If it was, I assigned the node the data values of that tetrahedron. If the point was outside of an object, I assigned a flag value to that particular node. By changing the number of nodes and their spacing, I could enhance the resolution of the final image.

When I started writing the program, my main problem was finding a quick way to see if a point was inside a tetrahedron. The method chosen uses a combination of dot product and cross. By taking the cross product of two edges that compose a face of the tetrahedron, I represented the entire face as a simple vector. The sign of the dot product of the vector of the face and the vector of the point tells me on which side of the face the point lies. However, this is order dependent; the signs are opposite if the data is declared in counter-clockwise manner rather than a clock-wise one. Because the data was unstructured, it was impossible to guarantee the proper order of the data without sorting it, and that would slow the program down too much. To fix this problem, I used the dot product of the face vector with the fourth point of tetrahedron as a guide to test the sign. If the two signs match, then the unknown point is on the correct side of the face. If this is true for all of the faces of the tetrahedron, then the point is indeed inside the tetrahedron.



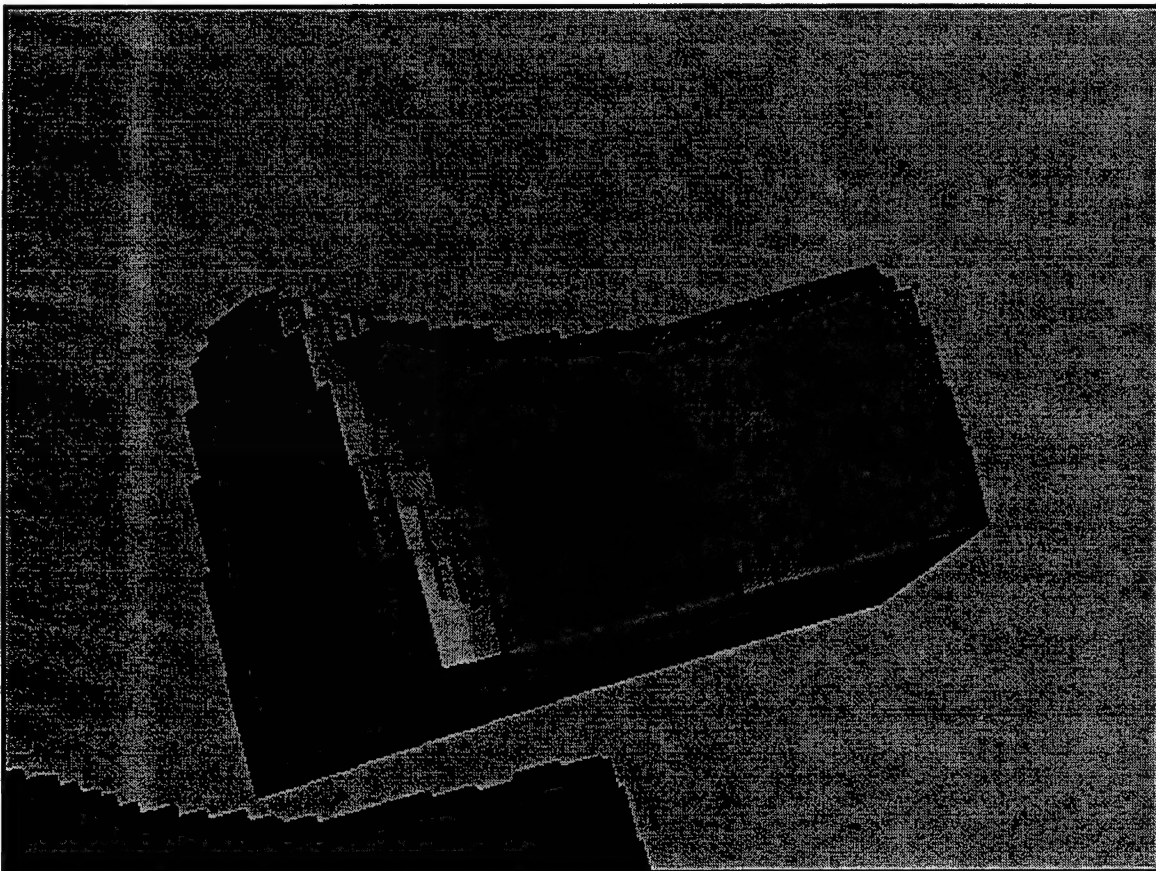
(Figure 53-3) Visual explanation of Cross Product and Dot Product.

Once the method was defined, the actual program was straightforward to write. However, computation time became a huge issue in program performance. After the first version of my program was finished, it took almost five hours to complete a data file with resolution of thirty increments in each direction. At higher levels of resolution, it would take almost 3 trillion calculations to calculate a data file.

Naturally, my mentor wanted to know if there was anyway to speed up this process, and accordingly I began to investigate different methods for converting the data. As I worked on the project, I optimized more and more of the program until a data file with resolution of one-hundred increments took only 35 seconds. The main method used to optimize the code was to preprocess the ranges of each tetrahedron and search through all points within the general range of a tetrahedron to see if they were inside a tetrahedron rather than searching through all of the tetrahedrons to find which a point was inside.

Another issue that became increasingly troublesome was the space requirements needed to store the output data files. As the resolution increased, the data files grew exponentially since the data was stored in the form of a cubic mesh. At a resolution of 100 increments, the data file was 100x100x100x64 bytes, or 64 megabytes. To compress the size of the data file, I found that some of the data I was including in the file was completely redundant. By removing the extraneous data, I was able to shrink the same data file to

16 megabytes. However, 16 megabytes is still large, especially for use on a PC. The next method I used was to change the format of the mesh that I used in my program design. Rather than using a cubic mesh, which contained a lot of empty space, I designed the program to 'fit' the mesh to the contours of the data file. Again this cut the size of the data file, now to around 10 megabytes. Finally, I investigate using file compression schemes to reduce the size. However, I was unable to find a program that would compress data on a UNIX Workstation, and be able to have that data decompressed on a PC. Because of this, I started designing data compression schemes to tie directly onto my program. Using a short program in AWK, I was able to cut the file size by a fifth. In the end, the original 64 megabyte data file was compressed to a little under two.



(Figure 53-4) Display of Slicer plot from the Tetrahedral Search at a medium level of resolution. This particular image is the temperature plot of a cylinder impact.

Along the course of the summer, I was also involved in many smaller projects. The first of these was a program to convert standard EPIC time-history data files into MS Excel's format so it could be easily plotted on a PC. This project became a study on different language application. While I originally started to write the program in FORTRAN, I found that the UNIX utility AWK was much more suited for such data manipulation. The program that does this task took almost sixty lines in FORTRAN, but I was able to do it in six in AWK. Also, the AWK program ran significantly faster than the compiled FORTRAN program, and naturally took up less space.

Another project that I accomplished was to convert EPIC material files, written as FORTRAN code, into Visual C++. I solved this as a two part project; First I wrote a FORTRAN program that included the material files and wrote the data into a text file. Secondly, I wrote an AWK program that converted the raw data into proper C++ syntax. I was happily surprised when the converted data was accepted without any errors.

Additionally, I worked with the other apprentices in MNMW to catalog a series of reel to reel films and VHS recordings. Finally, I helped design the MNMW homepages for use on the World Wide Web.

OBJECT ORIENTED DESIGN IN
HIERARCHICAL LEARNING SYSTEMS

Thomas E. Whalen

Northwestern University
606 University Place
Evanston, IL 60201

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

OBJECT ORIENTED DESIGN IN HIERARCHICAL LEARNING SYSTEMS

Thomas E. Whalen
Northwestern University

Abstract

Research in drive-reinforcement machine intelligence has progressed from single control systems to hierarchical networks of control systems, all the while improving and refining the learning mechanism. Each incremental research step in the past has been an individual event. The purpose of this project was to build a framework within which past research could be duplicated and future research could be facilitated. A common, object-oriented structure for network communication and construction was developed, allowing easy scaling of projects from single neuron models to large, complex networks. As a verification of the new system's reliability, past research was duplicated, resulting in some differences from the original behavior, but important similarities.

OBJECT ORIENTED DESIGN IN HIERARCHICAL LEARNING SYSTEMS

Thomas E. Whalen
Northwestern University

Introduction

In 1988, A. Harry Klopff published a paper in Psychobiology in which he developed a single neuron model which exhibited many properties of classical Pavlovian conditioning. The learning mechanism employed in this neuron he called "drive-reinforcement". In the journal Adaptive Behavior in 1993, Dr. Klopff, with James Morgan and Scott Weaver, expanded his original "drive-reinforcement" (DR) research to embrace hierarchical networks of neurons or control systems. A system made up of just 4 neurons was able to navigate though a multiple-T maze, exhibiting chaining of learned responses. In an accompanying paper, Leemon Baird and Harry Klopff detailed an improved version of the DR learning mechanism, called "modified drive-reinforcement".

In the future, DR will be refined and improved further, as new tests are performed to compare it to animal intelligence. This future research will require flexibility, as major changes in the adaptive mechanisms would significantly alter the structure of traditionally programmed simulations. An object-oriented framework, written in a powerful modern language like C++, however, could withstand these changes while providing a basic set of tested and accepted tools to build new adaptive systems and networks of systems.

Methodology

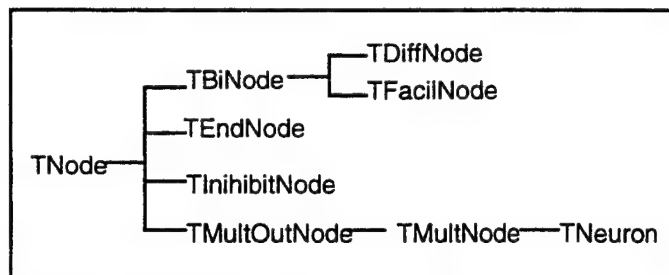
As part of the move from C and FORTRAN to C++, my mentor, Mark Sturgell, had coded the learning algorithm for original DR. He had implemented it on a "neuron" class of objects. My task was to develop structures to facilitate a network of neurons. This entailed describing the input-output relationships, various types of connections (direct and indirect) and deciding upon a network structure.

My first decision in developing the network structure was that simulations would be divided into two levels. The first, the "object" level would be made up of neuron and other network objects. It would handle the behavior of individual elements, such as neuronal learning and weighted summing of neuronal inputs. The "environment" level would be responsible for running the simulation, providing the input signal levels to the network, and timing the behavior of the learning system. The major question I faced was where to draw the line between the two layers. How much should be handled autonomously by the neurons, and how much should the individual implementation handle? I considered two solutions:

Hierarchical Model: In this model, the neurons would do little more than sum their inputs and learn. The environment would be responsible for directing the proper signals to the proper inputs, collecting outputs, and signaling the neurons to fire and learn. This would involve a very small object layer, but a highly complicated environmental layer, which would be very difficult to scale.

Feed-Forward Model: This model had the neurons supported by secondary network objects, which would handle complex issues like inhibition and facilitation of signals. This approach would have the environment layer responsible for sending the input signals into the sensors, and collecting the outputs from effectors, but all of the intermediate steps would be handled by the network itself. The object layer would be medium-sized, and the environmental layer could be as small as necessary to run the simulation. The neurons would operate independently from the environment, except for the initial inputs, the gathering of final outputs, and the timing of learning.

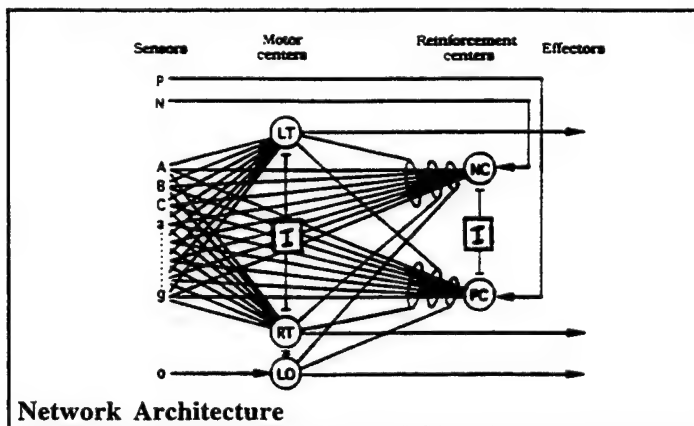
I decided to build the feed-forward model, because the implementation would be complex on the object level, but the individual implementations of simulations would be simpler to design.



Object Descendence Tree

The feed-forward model, as implemented, consisted of a simple class hierarchy for the basic network objects. As can be seen in the figure, the neuron class (TNeuron) was implemented as the most developed member of a family of network objects.

The root class was a generic network object known as a node (TNode). This class defined default input-output behavior for all objects in the network. The details of the implementation are contained in Appendix A.



All objects in the network have share certain abilities. The most important property they share is the capability to receive an input. This way, network objects can be connected in any order, because they can communicate under a standard. Here is an example of the input-output relationship between several objects in

the 1993 simulation:

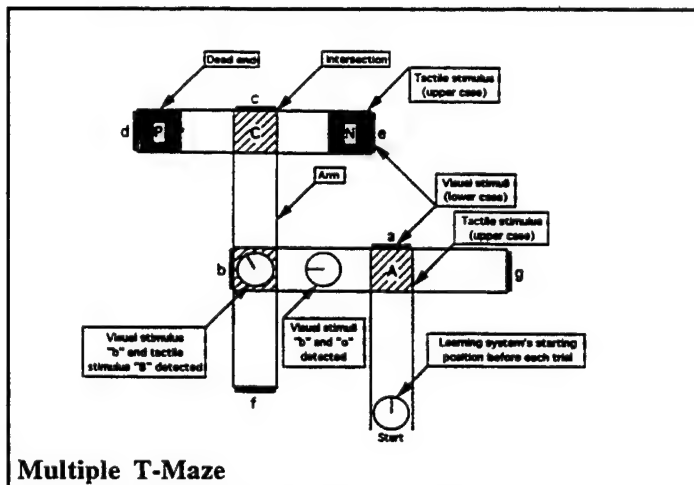
- Sensors a, b, c, d, e, f, g, A, B, and C (TNodes) receive their signal levels from the environmental level of the simulation
- The sensors output to neurons RT and LT (TNeurons), the motor turning centers. When each neuron receives its last input, it performs a weighted sum and sends that sum to the inhibitory node (TInhibitNode).
- Sensor o (TNode) receives its signal level from the environment and outputs to the LO motor center (TNeuron). The LO motor center is forced high if the o sensor is active. The LO motor center outputs to the inhibitory node.

- When the inhibitory node has received all of its inputs, it starts to do its work. The inhibitory node returns a true value to the neuron with the highest sum (RT, LT, or LO) and false values to all of the others. In the event of a tie, it returns false to all inputs.
- Motor centers receiving false values output zero, and the center which receives a true value outputs its calculated sum.

This is just a small part of the network's operations, but it illustrates how less complex network objects work in concert with the neuronal objects to simulate the learning system. In actuality, however, the motor centers are specialized sub-classes of TNeuron, an addition made necessary because of differences in learning methods.

Information on the various descendants of TNeuron is available in Appendix B.

Learning within the system presented a bit of a problem. Because the system was designed to be scaleable, from simulating a single neuron to an entire hierarchical network of neurons, the implementation of learning and timing had to be extremely flexible. Within the context of neural networks, learning is the process of calculating changes in the weights, and the application of those changes to the actual weights. In some simulations, the changes are calculated and applied every timestep. In others, the changes are summed across the entire trial and applied only at the end of the trial. This complication was not seen until the later stages of the development, and resulted in the division of timekeeping and learning into three separate methods. The first method incremented the neuron's internal time counter, using the current operating weights for the next timestep. The second accumulated changes in weights in a storage matrix, and the third method applied those changes to the operating weights. This allowed for an extremely flexible method of learning.



The completed learning system was tested according to the 1993 research. A multiple-T maze was simulated, and the learning system was tested within that environment. The T-Maze was implemented through a linked-list of objects which represented corridors and intersections. Each corridor knew where it led, and each intersection knew what was to the right and left. Each object also held

information on which sensors would be active, depending on which way the learning system was facing. This simple implementation was the quickest to code and test, although not very robust.

Results

Using the Klopff-Morgan-Weaver experimental results as a benchmark, I tested my implementation to confirm that it duplicated the results. I assigned several features of the older research as high-priority for duplication. These requirements for my learning system were:

- Learning system would chain behavior back from intersection C to intersection B to intersection A.
- Learning system would learn the maze in all mazes that the older implementation successfully competed.
- Learning system would not learn the maze in mazes where the older implementation failed.
- Learning system's behavior would seem rational, according to its equations.

My implementation met all of these requirements. Some other requirements my system met in part were:

- Learning system would duplicate the behavior of the original implementation.

- Learning system would learn at the same rate as the original implementation.
- Learning system would exhibit numerical equivalence to the original implementation in weights and outputs.

(This requirement was discovered to be unverifiable.)

I discovered that my learning system implementation did not **exactly** duplicate the original research, but to have expected it to would have been overly optimistic. The source code for the original was unavailable, and reverse-engineering the system from the published paper resulted in some ambiguity. Additionally, the new research was performed on a different computer than the old, written in a different language, and in a completely new way.

Because of these possible explanations for the deviation from the original behavior, and because the system met all of the major requirements for duplication, my research team (including the authors of the original paper) decided the differences were incidental.

References

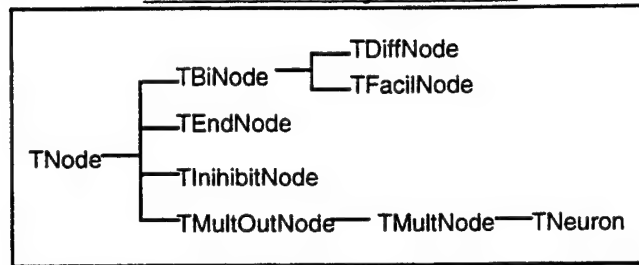
Baird, L. C. & Klopff, A. H. (1993). A Hierarchical Network of Provably Optimal Learning Control Systems: Extensions of the Associative Control Process (ACP) Network. *Adaptive Behavior*. 1, 321-352.

Klopff, A. H. (1988). A neuronal model of classical conditioning. *Psychobiology*, 16, 85-125.

Klopff, A.H., Morgan, J.S. & Weaver, S. E. (1993). A Hierarchical Network of Control Systems that Learn: Modeling Nervous System Function During Classical and Instrumental Conditioning. *Adaptive Behavior*. 1, 263-319.

Appendix A:

Basic Network Objects Manual

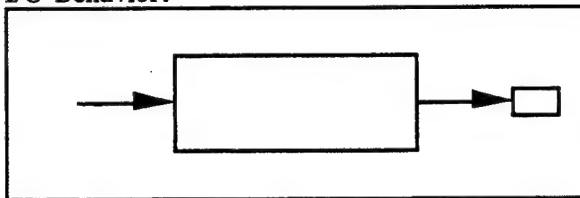


TNode

Description:

TNode is the base class for all elements in a network. The main reason for having this base class is so that the programmer can be sure that all objects in the network will share certain basic behaviors relating to I/O.

I/O Behavior:



A **TNode** takes one input, which can be connected from anything, and gives one output, which must be another object of type **TNode**. **TNode** performs no processing on the value passed.

Public Methods:

ConnectIn(TNode pointer) is provided as a null function on the **TNode** class. It is designed to be overloaded by descended classes.

ConnectOut(TNode pointer) sets the *dest* Data Member on the object. This function is used to set up an output relationship -- the **TNode** will output to the object pointed to.

Input(float, [TNode pointer]) receives input from one source, does no processing on the value and passes the value along to its destination set by ***ConnectOut()***. The second argument, a pointer to a **TNode** is optional. This argument is used in descendent classes to discern the origin of the input.

Public Data Members:

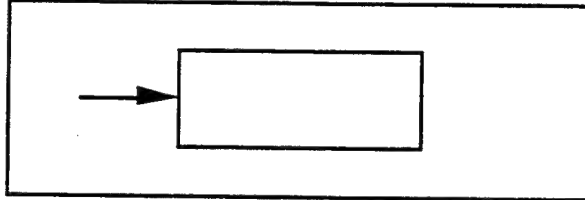
y and *yLast* are designed to hold the value passed through the **TNode**. *y* holds the most recent, while *yLast* holds the one before that.

TEndNode

Description:

TEndNode is directly derived from **TNode**. Its main function is to serve as a connection to the outside environment.

I/O Behavior:



Instead of the usual I/O behaviors, a **TEndNode** accepts input from anything, but does no output, allowing a branch of the network to terminate. The environment would then retrieve the value from the *y* data member.

Public Methods: (None unique)

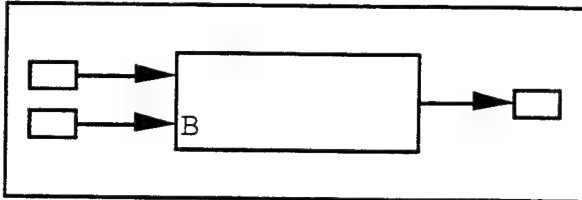
Public Data Members: (None unique)

TBiNode

Description:

TBiNode is a base class descended from **TNode**, not designed to be actually used. It handles two distinct input streams, allowing derived classes to treat the signals differently. Non-commutative operations like subtraction, division, and some logical comparisons would need this input typing.

I/O Behavior:



TBiNode is set up to accept two separate and distinct signals from **TNodes** on *Input()*, and output to one **TNode**. Only when it has received both signals will it output. Note that as coded on **TBiNode**, no input can be passed though -- the processing of the two inputs must be implemented on descendant classes.

Public Methods:

ConnectIn(TNode pointer) and *ConnectInB(TNode pointer)* implement the first part of **TBiNode**'s structure. They store pointers to the objects which will be sending input. This allows the **TBiNode** to know which object is sending what input. *ConnectInB()* sets up the secondary input stream, as opposed to the regular *ConnectIn()*.

Input(float, TNode pointer) overloads **TNode** in order to set up the wait for both inputs and to handle the differentiation between them. Notice the pointer is no longer optional -- **TBiNodes** can only accept input from other **TNodes** -- they must know where the input is coming from.

Public Data Members: (None unique)

TDiffNode

Description:

TDiffNode is one implementation of a **TBiNode**. Its processing is set up to subtract the secondary (B) signal from the primary signal.

I/O Behavior: (Inherited from **TBiNode**)

Public Methods: (None unique)

Public Data Members: (None unique)

TFacilNode

Description:

TFacilNode is another implementation of a **TBiNode**. It is designed to duplicate the action of both a facilitatory and facilitated connection. The primary signal is considered the facilitated signal, and the secondary (B) signal is the facilitator. The processing is set up so that if the B signal is non-zero, the primary signal is passed though. Otherwise, the node outputs zero.

I/O Behavior: (Inherited from **TBiNode**)

Public Methods: (None unique)

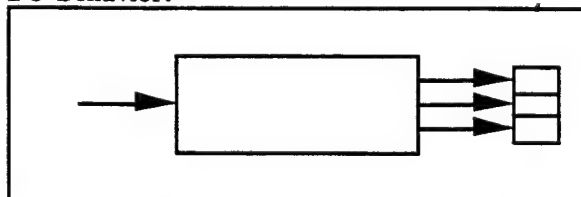
Public Data Members: (None unique)

TMultOutNode

Description:

A **TMultOutNode** is a value broadcaster -- valuable for emulating a sensor that must report values to multiple neurons, for example.

I/O Behavior:



Accepting one input from anything, a **TMultOutNode** sends that value along to multiple **TNodes**. It does not change the value in any way, merely passes it out to the *Input()* methods of its destinations.

Public Methods:

ConnectOut(TNode pointer) handles multiple outputs, so the program would call it once for each output to be connected.

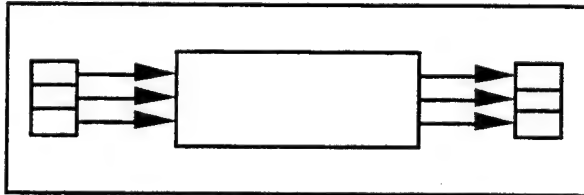
Public Data Members: (None Unique)

TMultNode

Description:

A **TMultNode** is an intermediate class in the transition between a **TNode** and a **TNeuron**. It handles both multiple inputs and multiple outputs, but lacks the processing between input and output to make it useful as a parent class.

I/O Behavior:



Accepting multiple inputs from **TNodes** on *Input()*, the **TMultNode** waits until it has received all of the inputs it is expecting, then outputs a value (determined internally) to all of the connected outputs.

Public Methods:

ConnectIn(TNode pointer) takes a pointer to a **TNode** as an argument. The **TMultNode** needs to know where each of its inputs is coming from, in order to keep them separate.

Input(float, TNode pointer) stores the floating point value it receives. The pointer in the argument list indicates the origin of the value, allowing the **TMultNode** to keep its multiple inputs separate. When the **TMultNode** has received all of the inputs it expects, it will process and output.

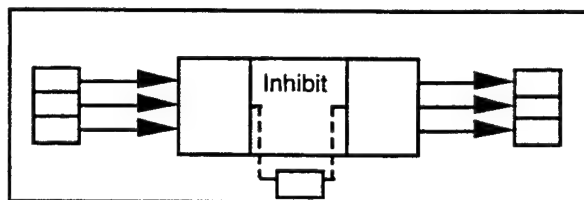
Public Data Members: (None Unique)

TNeuron

Description:

A **TNeuron** is the basic Neuron in this object hierarchy. It is a direct descendant from **TMultNode**. To the structure of **TMultNode**, it adds a weighted summing process in between input and output, and provides a basic framework for modifying input weight factors. Descendant classes should implement specific learning algorithms.

I/O Behavior:



This behavior is very much like a **TMultNode**, except for the weighted summing process and what occurs immediately after: If the neuron is inhibited, the **TNeuron** outputs to a **TInhibitNode**, which later sends a Boolean signal to the **TNeuron**, the truth of which determines whether the **TNeuron** outputs zero or its regular output

through normal **TMultNode** channels. If the **TNeuron** is not inhibited, I/O proceeds normally.

Public Methods:

ConnectIn(TNode pointer, [Boolean]) supersedes **ConnectIn()** on **TMultNode**, with the pointer specifying the node to be connected in, but with a second argument. The optional Boolean argument indicates if the connection is plastic or not. A false value passed in indicates a non-plastic (hardwired) connection. A true value, or no value at all indicates a plastic connection.

InhibitWith(TInhibitNode pointer) sets up the neuron to be inhibited. The pointer indicates a **TInhibitNode**, a node which handles the actual inhibition.

Inhibit(short) is called with a true-false value by the inhibiting node to indicate to the neuron whether it should output zero or its sum. **Inhibit()** actually causes the output function to run.

Input(float, TNode pointer) supersedes **TMultNode** in order to implement the inhibition. It behaves regularly, but does not call the output function if the node is inhibited.

AccumWeights() handles the basic learning algorithm, which on this base class is pretty much null. On descended classes, delta-weights are calculated and added to the **dWPos** and **dWNeg** matrices.

UpdateWeights() adds the delta-weight matrices to the operating weight matrices. This basically means that weight changes only start to take effect after this method is called.

Next_t() moves the operating weights into their proper places in the history matrices and increments the neuron's internal timestep counter. This method should be called every timestep, regardless of how learning takes place in the particular experiment.

Public Data Members:

wPos and **wNeg** are the weight matrices which are actually used to calculate the weighted sum in the neuron's I/O process. They are dimensioned by the number of inputs. The first is the excitatory weight component, and the second is the inhibitory.

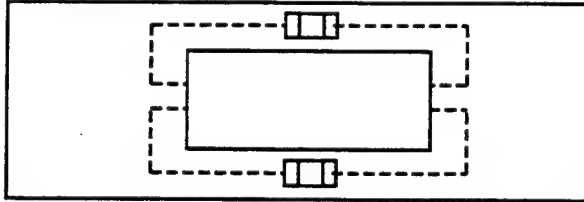
dWPos and **dWNeg** are the matrices which store the changes in the weights, created by the **AccumWeights()** method. They have as many slots as the neuron has inputs.

TInhibitNode

Description:

A **TInhibitNode** works hand-in-hand with an inhibited **TNeuron**. Basically, it figures out which of its inputs is the greatest and tells all of its sources whether they were it or not.

I/O Behavior:



Accepting multiple inputs on *Input()*, the **TInhibitNode** calculates the highest signal and sends a true value to that source's *Inhibit()* method. All other sources get false values. In the event of a tie, all the sources get false values. Also, the node can be set up to inhibit all of its sources for one input cycle. When all of the inputs have

been received, the **TInhibitNode** does its thing.

Public Methods:

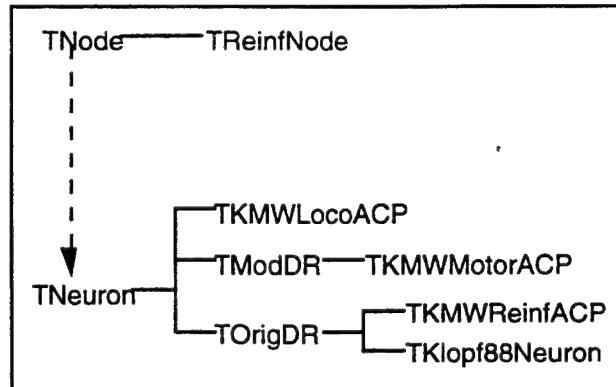
ConnectIn(TNeuron pointer) overloads **TNode** to require that the input be a **TNeuron**. It also maintains an array of pointers to the sources, to identify the streams.

Input(float, TNeuron pointer) handles the comparison process and accepts more than one input sources.

InhibitAll() instructs the node to inhibit all of its sources for one cycle.

Public Data Members: (None Unique)

Appendix B:
Neuron Object Type Manual



TOrigDR

Description:

The **TOrigDR** class contains the algorithms necessary to implement 'original' drive-reinforcement learning.

Research Reference:

Klopf, A. H. (1988). A neuronal model of classical conditioning. *Psychobiology*, **16**, 85-125.

Learning Algorithm:

$$\Delta w_i = \Delta y(t) \sum_{j=1}^{\tau} c_j |w_i(t-j)| \Delta x_i(t-j)$$

TModDR

Description:

The **TModDR** contains the algorithms necessary to implement 'modified' drive-reinforcement learning. However, this implementation is untested and likely to need minimal debugging.

Research Reference:

Baird, L. C. & Klopf, A. H. (1993). A Hierarchical Network of Provably Optimal Learning Control Systems: Extensions of the Associative Control Process (ACP) Network. *Adaptive Behavior*, **1**, 321-352.

Learning Algorithm:

$$\Delta w_i(t) = [\gamma y(t) - y(t-1) + R(t)] \times \sum_{j=1}^{\tau} c_j |w_i(t-j)| [\Delta x_i(t-j)]^+$$

Public Methods:

Reinf(float) accepts the reinforcement signal ("R(t)" in the equation above). As of this writing, the neuron only accepts one signal, so any summing or subtraction has to be done in another node.

TReinfNode**Description:**

The **TReinfNode** is designed to help implement modified reinforcement learning. Often in ACP networks, the output of one ACP must be used as a reinforcing signal for another ACP. The **TReinfNode** allows this conversion to occur. **TReinfNode** is a child of **TNode**.

I/O Behavior:

A **TReinfNode** takes one input, which can be connected from anything, and gives one output, which must be connected to a **TModDR** object. Instead of following normal I/O methods, the value is passed to the neuron's *Reinf()* method.

Public Methods:

ConnectOut(TModDR pointer) supercedes **TNode** in order to force the connection to be to an object with a *Reinf()* method to call.

TKlopf88Neuron**Description:**

The **TKlopf88Neuron** implements the structures used in the single-neuron simulation in the research below. This test of original drive-reinforcement placed a single, uninhibited neuron in a test environment.

Research Reference:

Klopf, A. H. (1988). A neuronal model of classical conditioning. *Psychobiology*, **16**, 85-125.

Learning Algorithm:

$$\Delta w_i = \Delta y(t) \sum_{j=1}^{\tau} c_j |w_i(t-j)| \Delta x_i(t-j)$$

TKMWMotorAcp

Description:

The **TKMWMotorACP** duplicates the behavior of the turning motor centers in the research below. Although the learning method is different, (a kludge) the motor centers use **TModDR** as a parent class to gain the *Reinf()* functionality, a part of motor learning.

Research Reference:

Klopf, A.H., Morgan, J.S. & Weaver, S. E. (1993). A Hierarchical Network of Control Systems that Learn: Modeling Nervous System Function During Classical and Instrumental Conditioning. *Adaptive Behavior*. 1, 263-319.

Learning Algorithm:

$$\Delta w_i(t) = [c][|w_i(t)|][\Delta x_i(t)]^+ [y_p(t) - y_n(t) - y(t)]\{sgn[y(t)]\}$$

TKMWLocoACP

Description:

The **TKMWLocoACP** duplicates the behavior of the locomotion motor center in the research below. The locomotion center does not learn, but needs to be inhibited, so the base class used was **TNeuron**. In the research, if the learning system could go forward, it was forced to. To implement this, the output of the motor center was limited to the values 0 and 2. If the motor center as on at all, it had a value of 2. The maximum output of the other motor centers was 1, so if the locomotion center was active, it was the strongest signal, and thus inhibited all the others.

Research Reference:

Klopf, A.H., Morgan, J.S. & Weaver, S. E. (1993). A Hierarchical Network of Control Systems that Learn: Modeling Nervous System Function During Classical and Instrumental Conditioning. *Adaptive Behavior*. 1, 263-319.

TKMWReinfACP

Description:

In the research below, there were two reinforcement centers. **TKMWReinfACP** models these two centers. Because the two centers used an only slightly modified version of original DR learning, the parent class used was **TOrigDR**.

Research Reference:

Klopf, A.H., Morgan, J.S. & Weaver, S. E. (1993). A Hierarchical Network of Control Systems that Learn: Modeling Nervous System Function During Classical and Instrumental Conditioning. *Adaptive Behavior*. **1**, 263-319.

Learning Algorithm:

$$\Delta w_i = \Delta y(t) \sum_{j=1}^{\tau} c_j |w_i(t-j)| [\Delta x_i(t-j)]^+$$

Translator for Epic FEM Time-History Files

Gabrielle White Wolf

Choctawhatchee High School
110 Racetrack Rd.
Ft. Walton Beach, FL 32548

Final Report for:
High School Apprentice Program
Wright Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Laboratory

August 1995

Translator for EPIC FEM Time-History Files

Gabrielle White Wolf
Choctawhatchee High School

Abstract

The EPIC hydrocode records highly dynamic variables such as temperature or pressure over a period of time at certain locations on an object; it transfers this data to an output time-history file in binary format, which is created each time the EPIC program is run. The EPIC POST2 time-history post-processor can convert this to ASCII format to make it portable, but it is not in a format that is easy to access via most commercial X-Y plotters such as Microsoft Excel and Spyglass Plot software. This translator performs the conversion so that the data is accessible to these programs.

The translator program was created in FORTRAN in order to sort the data into system, node and element information and write the data to specific files for further use. The translator was run on several example files from EPIC software and the output files of data were then graphed using Spyglass Plot; results were compared to the example graphs included with the EPIC program. The extreme similarity between the two graphs implied that the translator could be used for further sorting of data in time-history files for graphing and analysis.

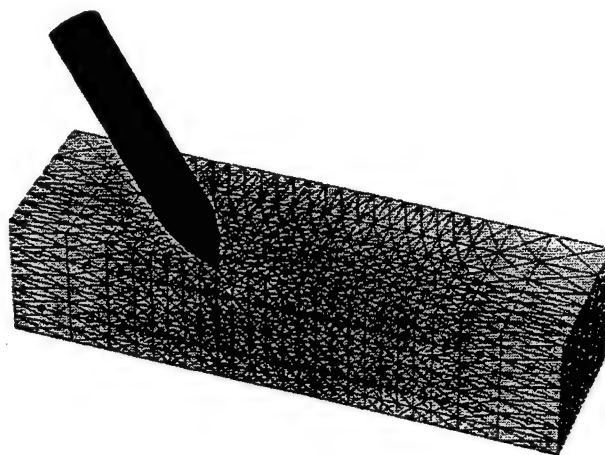
Translator for EPIC FEM Time-History Files

Gabrielle White Wolf

Introduction

The EPIC hydrocode, a time-dependent finite element code, was developed by Dr. Gordon Johnson at Allian Technology Systems under the sponsorship of Wright Labs. This code simulates highly dynamic events, such as warhead formations, and is used to display computational variables, such as stress or pressure, over a certain period of time.

Generally, EPIC95 recognizes three kinds of data points: system, element or node points. Figure 57-1 is an image generated by EPIC, where a penetrator is entering a target---in this example, both the penetrator and the target would be considered system points. Occasionally, just the penetrator or just the target is being examined and in such situations, these are determined to be chunk data, or subsets of system data. System data is made up of individual tetrahedrons and these are considered element data, while the vertices on the tetrahedrons are the node data points. The variables examined on these particular points are represented by equations that are integrated over time to determine the values.



(Figure 57-1)

The output file of an EPIC run begins with four numbers that act as identifiers for that file; these include two revisions numbers and another that labels a particular simulation run. The format of the output file follows a specific structure shown in figure 57-2---following the initial set of numbers are a triplet of numbers, one of which contains an identifier that specifies which type of data is being sampled. It is the second number of this triplet that signals the type, "one" being a system point, "two" being node points and "three" being element points. Following this triplet is an identifying label for that particular point, and the next block of numbers is corresponding data for that point at that particular moment in time.

6	6	622	3
3	1	2	
0			
0.00E+00	23576	0.00E+00	0
0.00E+00	0	0.00E+00	0
0.00E+00	0	0.00E+00	0
0.00E+00			
3	2	2	
11			
0.00E+00	33056	3.00E-06	0
0.00E+00	0	0.00E+00	0
0.00E+00	33630	1.00E+03	0
0.00E+00			
3	3	2	
120			
0.00E+00	23576	0.00E+00	0
0.00E+00	0	0.00E+00	0
0.00E+00	22609	4.00E+01	0
0.00E+00	0	0.00E+00	0

(Figure 57-2)

The output file lists every system, node and element point and its corresponding data at a certain time interval, then increments the time and begins to list every point and its new set of data at the new interval of time.

Equipment Used

- **FORTTRAN Programming Language:** FORTRAN Version 77 was used to create the translator program for the time-history data files.
- **EPIC95:** The EPIC hydrocode created the output time-history data files that were examined and sorted.
- **Spyglass Plot:** This software can import text files, arrays and output files and transfer them into readable form; the data can then be plotted and graphed as desired using parameters available on the program.
- **Silicon graphics 150 MHz Indy Workstation**
- **Gateway 2000 40DX2-66 PC**

Methodology

FORTTRAN was the programming language used to create the translator program for the EPIC time-history files; the program consisted of three major parts---first the number of system, node and element points present in the output file were determined, next data was stored in appropriate arrays, and finally the arrays were written to files with names based on their contents.

As the different points occurred in a pattern in the input file, the program had to determine how many different types of points were present in the file. Primarily, the program read in the first triplet and depending on the value of the second number in the triplet, sent the following number (the identifying labeler) to an array. All system points sent the values of the label to "sys(i)", node points to "node(j)", and element points to "elem(k)"---the values of "i", "j" and "k" served as counters to determine the number of points, and the program included a checkpoint to determine when data was repeated. At this point, the

program both printed to the screen and stored the values of "i", "j", and "k", the total number of system, element and node points present in the output file, and the input file was rewound to the beginning.

The second section of the program read the data following the triplet and label and stored in the appropriate arrays based on the middle variable of the triplet. The data is stored into either "datas" (the system array), "datan" (the node array), or "datae" (the element array). The program uses the "i, j, and k" values as loop counters; this process is repeated for each type of data and an error stipulation is entered to send the program to the third section in the event of an error or the end of the file is reached.

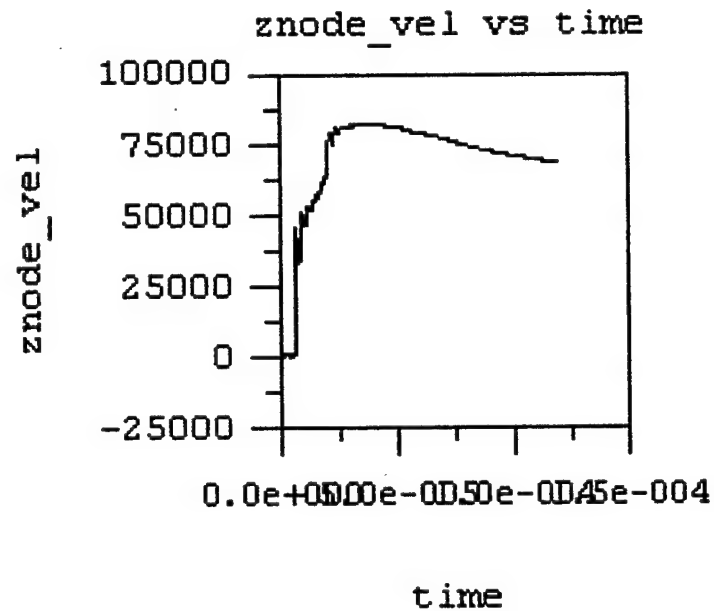
Finally in the third section of the translator, each of the arrays are written to individual and specific files. All the data in the system array, "datas", was written to files named "system" plus the label number specific to that system point, accessed from the initial "sys(i)" array. Similarly, the data from "datan" and "datae" was written to files named "node" or "element" and followed by the corresponding label number.

Spyglass Plot was then used to open each individual file and examine the results graphically; Spyglass reads in the files and the user can plot column against column and set specifications within the program.

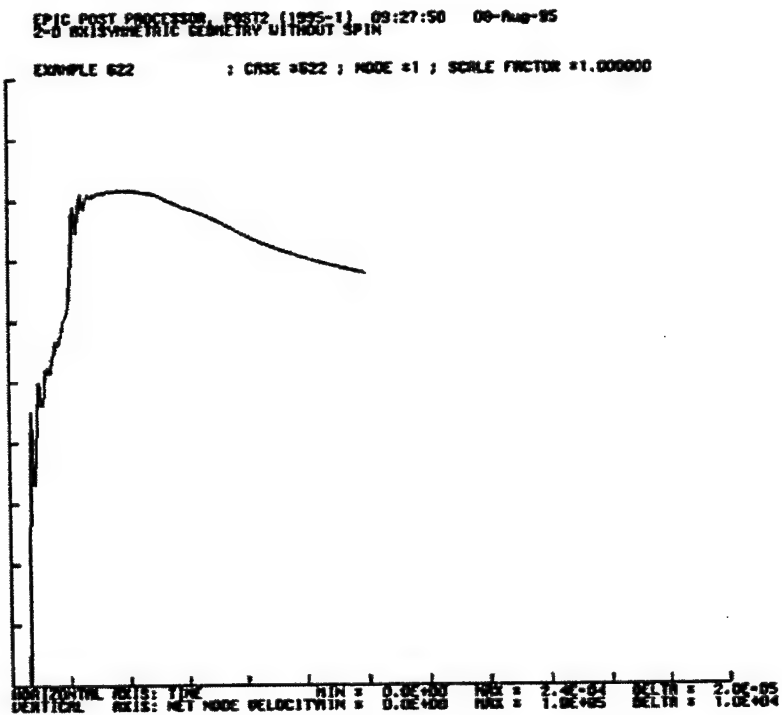
To examine the accuracy and efficiency of the translator program, the program was run on an example file included with EPIC; the program produced five files, including "sys00000", "nod00011", and "elem00120"; the plots created on the Spyglass Software were then compared to EPIC outputs to verify the results.

Results

A translator for EPIC time-history data files was created and results were verified using other plotting techniques. Also, several test problems from EPIC were run and animated on an SGI workstation.



SPYGLASS PLOT GRAPH
 NODE POINT "00011"
 Z-NODAL VELOCITY VS. TIME



EPIC GRAPH
 NODE POINT "00011"
 Z-NODAL VELOCITY VS. TIME

Additional Accomplishments

As well as developing basic skills in FORTRAN Programming Language, and being exposed to EPIC and Spyglass software, other efficient FORTRAN programs were created. Also, several co-workers presented tasks to be completed, such as cataloging reel to reel films and VHS videos and organizing them into different categories for storage.

Another major accomplishment during the summer was the creation of movies in EPIC95. The EPIC code was run for 25 test problems and images were displayed and captured; these images were stored as RGB files in sequential order. The individual images then had to be compressed into a movie, with the use of the "makemovie" function on the Silicon Graphics workstation---the images were compressed into movies and stored into a separate directory for further use.

Acknowledgments

This summer I would like to acknowledge the following people . . .

- *Michael E Nixon
- *Mr. Deiler and Mr. Harrison
- *Major Howard Gans
- *Computational Mechanics
- *High School Apprentices

TRANSLATOR PROGRAM:

- * The translator program reads output files from EPIC runs and writes
- * system, element, and node data to corresponding files based on the type
- * of point examined and its identifying label.

```
Program Translator
Integer a,b,c,sys(50),elem(50),node(50)
Integer ts,tn,te,isis,ielem,inod
character*15 filenames, filenameen,filenameee
Dimension datas(100,85,100),datan(100,16,3600)
Dimension datae(100,18,1200)
ts=1
tn=1
te=1
i=0
j=0
k=0
Open (Unit=25, File='th.in')
```

- * Format statements:

```
10 Format (3I5)
20 Format (16(e12.6,1x))
21 Format (18(e12.6,1x))
22 Format (82(e12.6,1x))
30 Format (4I5)
11 Format (A10)
9 Format (I5)
```

- * The first four numbers of the input file are
read (25,*) i1, i2, i3, i4

- * The value of 'b' will determine which type of point is being examined.
1 Read (25,10) a,b,c

- * System data:

```
if (b.eq.1) then
  i=i+1
  read (25,*) sys(i)
  if (i.gt.1) then
    if (sys(i).eq.sys(1)) then
      i = i-1
    go to 4
  endif
endif
Read (25,*) (datas(i,m,1),m=1,5)
Read (25,*) (datas(i,m,1),m=6,10)
Read (25,*) (datas(i,m,1),m=11,15)
Read (25,*) (datas(i,m,1),m=16,20)
Read (25,*) (datas(i,m,1),m=21,25)
Read (25,*) (datas(i,m,1),m=26,30)
```

```

Read (25,*) (datas(i,m,1),m=31,35)
Read (25,*) (datas(i,m,1),m=36,40)
Read (25,*) (datas(i,m,1),m=41,45)
Read (25,*) (datas(i,m,1),m=46,50)
Read (25,*) (datas(i,m,1),m=51,55)
Read (25,*) (datas(i,m,1),m=56,60)
Read (25,*) (datas(i,m,1),m=61,65)
Read (25,*) (datas(i,m,1),m=66,70)
Read (25,*) (datas(i,m,1),m=71,75)
Read (25,*) (datas(i,m,1),m=76,80)
Read (25,*) (datas(i,m,1),m=81,82)

```

* Node data:

```

elseif (b.eq.2) then
  j=j+1
  read (25,*) node(j)
  if (j.gt.1) then
    if (node(j).eq.node(1)) then
      j = j-1
    go to 4
  endif
endif
Read (25,*) (datan(j,m,1),m=1,5)
read (25,*) (datan(j,m,1),m=6,10)
read (25,*) (datan(j,m,1),m=11,15)
read (25,*) datan(j,16,1)

```

* Element data:

```

elseif (b.eq.3) then
  k=k+1
  read (25,*) elem(k)
  if (k.gt.1) then
    if (elem(k).eq.elem(1)) then
      k = k-1
    go to 4
  endif
endif
read (25,*) (datae(k,m,1),m=1,5)
read (25,*) (datae(k,m,1),m=6,10)
read (25,*) (datae(k,m,1),m=11,15)
read (25,*) (datae(k,m,1),m=16,18)

```

```

endif
go to 1

```

4 rewind 25

* 'i, j, and k' represent the number of system, node and element points
 * in the input file.

```

write(*,*)'i,j,k = ',i,j,k

```

* The second part of the program reads in all data to arrays, depending
 * again on the value of 'b', and type of point being examined. Once the
 * end of the file is reached, or an error occurs, the program is directed
 * to line 3.

```
2 Read (25,10,end=3,err=3) a,b,c
```

* System data---'datas(x,y,z)' stores in all system data. 'is' is the
 * number of system points, 'm' is location, and 'ts' is the number of
 * times system points are sampled.

```
if (b.eq.1) then
  isys=isys+1
  do is=1,i
    read (25,*) msys
    Read (25,*) (datas (is,m,ts),m=1,5)
    read (25,*) (datas (is,m,ts),m=6,10)
    read (25,*) (datas (is,m,ts),m=11,15)
    read (25,*) (datas (is,m,ts),m=16,20)
    read (25,*) (datas (is,m,ts),m=21,25)
    read (25,*) (datas (is,m,ts),m=26,30)
    read (25,*) (datas (is,m,ts),m=31,35)
    read (25,*) (datas (is,m,ts),m=36,40)
    read (25,*) (datas (is,m,ts),m=41,45)
    read (25,*) (datas (is,m,ts),m=46,50)
    Read (25,*) (datas (is,m,ts),m=51,55)
    Read (25,*) (datas (is,m,ts),m=56,60)
    Read (25,*) (datas (is,m,ts),m=61,65)
    Read (25,*) (datas(is,m,ts),m=66,70)
    Read (25,*) (datas(is,m,ts),m=71,75)
    Read (25,*) (datas(is,m,ts),m=76,80)
    Read (25,*) (datas(is,m,ts),m=81,82)
  enddo
  ts=ts+1
```

* Node data---'datan(x,y,z)' stores all node data. 'in' is the
 * number of node points present, 'm' is location, 'tn' is number of
 * times node points are sampled.

```
elseif (b.eq.2) then
  do in=1,j
    inod=inod+1
    read (25,*) mnod
    Read (25,*) (datan (in,m,tn),m=1,5)
    read (25,*) (datan (in,m,tn),m=6,10)
    read (25,*) (datan (in,m,tn),m=11,15)
    read (25,*) datan (in,16,tn)
  enddo
  tn=tn+1
```

* Element data---'datae(x,y,z)' stores all element data. 'ie' is the
 * number of element points present, 'm' is location, 'tn' is number of

```

* times element points are sampled.
elseif (b.eq.3) then
  ielem=ielem+1
  do ie=i,k
    read (25,*) mele
    Read (25,*) (datae (ie,m,te),m=1,5)
    read (25,*) (datae (ie,m,te),m=6,10)
    read (25,*) (datae (ie,m,te),m=11,15)
    read (25,*) (datae (ie,m,te),m=16,18)
  enddo
  te=te+1

```

```

endif

```

```

go to 2

```

```

3 ts=ts-1
  tn=tn-1
  te=te-1

```

```

write(*,*)'ts = ',ts
write(*,*)'tn = ',tn
write(*,*)'te = ',te
write(*,*)'number of sys=',isys
write(*,*)'number of nodes=',inod
write(*,*)'number of elements=',ielem

```

*The third part of the program writes each of the arrays created in
 *second part to specific files based on the label of each points.

*System data---read from 'datas' to 'filenames', which is sys+label
 *number.

```

if (i.ne.0) then
  do is=1,i
    write(filenames,'(a3,i5.5,1x)')sys',sys(is)
    open(20,file=filenames)
    write(20,*)'time tot_ene kin_ene int_ene plas_eng xmaxcoor
1 ymaxcoor zmaxcoor xmincoor ymincoor zmincoor xcr_grav ycr_grav
1 zcr_grav xlin_mom ylin_mom zlin_mom xvel yvel zvel netvel
1 xang_mom yang_mom zang_mom xang_vel yang_vel zang_vel mass
1 xacc yacc zacc'
    do it=1,ts
      write (20,22) (datas(is,m,it),m=1,82)
    enddo
  enddo
endif

```

*Node data---read from 'datan' to 'filamen', which is node+node
 *label.


```

if (j.ne.0) then
  do in=1,j
    write(filename,'(a3,i5.5,1x)')'nod',node(in)
    open(40,file=filename)
    write(40,*) 'time xnode_pos ynode_pos znode_pos xnode_vel
1 ynode_vel znode_vel net_vel xnode_acc ynode_acc znode_acc
1 nod_press xforce yforce zforce node_temp'
    do jt=1,tn
      write (40,20) (datan(in,m,jt),m=1,16)
    enddo
  enddo
endif

```

*Element data---read from 'datae' to 'filenamee', which is element+
 *element label.

```

if (k.ne.0) then
  do ie=1,k
    write(filenamee,'(a3,i5.5,1x)')'ele',elem(ie)
    open(60,file=filenamee)
    write(60,*) 'time ele_pres ele_prs(na) vonm_strs eqplastrn
1 dam_burn temp plas int_ene log_strn xnorm_strs ynorm_strs
1 znorm_strs xy_shr xz_shr zy_shr ratio_mn/vm volum_strn'
    do kt=1,te
      write (60,21) (datae(ie,m,kt),m=1,18)
    enddo
  enddo
endif

stop
end

```

**MICROSTRUCTURAL ANALYSIS OF PBXN-110 TO DETERMINE
VOID VOLUME FRACTION AND PARTICLE DISTRIBUTION**

Tuan P. Yang

**Choctawhatchee High School
110 Racetrack Road
Fort Walton Beach, FL
32548**

**Final Report for:
High School Apprentice Program
Wright Laboratory**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC**

and

Wright Laboratory

August 1995

MICROSTRUCTURAL ANALYSIS OF PBXN-110 TO DETERMINE VOID VOLUME FRACTION AND PARTICLE ANALYSIS

Tuan P. Yang

Abstract

The explosive, PBXN-110 and its solid components, HMX Class II and III, were studied. PBXN-110 is composed of sixty-six percent HMX Class III crystals, twenty-two percent HMX Class II crystals, and twelve percent inert binder. Particle analyses of HMX crystals show that the majority of the sizes range from three-hundred microns up to eight-hundred and forty-one microns. Scanning electron microscopy (SEM) confirmed the measurements and showed few flaws in the crystals themselves. Mixing and casting were carried out under vacuum. The different samples were then cut up and densities determined. Three different methods were used to determine the densities of the PBX specimen. These included: the penta-pycnometer apparatus, the Mettler Density Determination Apparatus and hand calculations using weight, length, and diameter. Then SEM pictures were taken to determine the topography of the PBX. Using these pictures, physical measurements were accomplished and comparisons were made. Results show that the particle distribution does not match with those of the HMX specifications and that the void volume appears to be around the crystals and not distributed throughout the rubber matrix. Results also showed that the SEM analysis does provide quantification of the HMX particles in the matrix and can also be numerically modeled.

MICROSTRUCTURAL ANALYSIS OF PBXN-110 TO DETERMINE VOID VOLUME FRACTION AND PARTICLE DISTRIBUTION

Tuan P. Yang

Introduction

In support of the "Mechanical Properties of Explosives under High Strain Rate conditions" program, a micro - level investigation has been on-going to understand how the explosive matrix affects the overall behavior of the explosive under such conditions. The explosive components of any given warhead must be studied first in order to understand how it will respond to external strain stimuli. For this investigation, PBXN-110 was chosen due to its compositional materials and elasticity which will work well for post impact analysis.

PBXN-110 is composed of eighty-eight percent HMX crystals and a rubbery binder. A particle analysis will be carried out on the crystals and thus will quantify how the HMX crystals and voids are distributed throughout the vacuum and cast samples. SEM analysis will also be performed in order to study the internal matrix and to see if it is possible to numerically quantify the specimen.

Methodology

The method of making PBXN-110 consists of mixing together Classes II and III HMX crystals with an inert rubber binder. Prior to the mixing operation a particle analysis was carried out on the class III HMX crystals. To do this, a sieve analysis was used to distinguish the particle sizes.

This process was carried out using a series of sieves stacked on top of one another with the bottom container being a catch pan. Each sieve is composed of a pan with a wire mesh bottom. The size of the openings in the wire mesh get progressively smaller with each pan going from top to bottom. One hundred grams of HMX was added to the top pan and the entire stack vibrated for a set amount of time. In this case, it was for ten minutes. After the set time had expired, it was left for five more minutes in

order to allow the particles and dust to settle as a safety precaution. The particles in each pan were then weighed and the weight percent established for each particle size range.

The HMX crystals were then analyzed by SEM to determine the particle size distribution by an independent method and to distinguish the crystal's morphology, flaws, and distribution.

Some representative samples of crystals were glued to a mounting and sputter coated with gold, so the electron beam could focus properly on the particles. Comparisons were then made to the HMX particle specifications.

For the mixing process, a half-pint size Baker Perkins High-Shear Vertical Mixer was used. Five samples of one hundred and fifty grams of PBXN-110 were processed. Each sample was mixed under vacuum in fifteen minute cycles. In this particular mix, there were five cycles. The first cycle consisted of mixing the rubber binder. This was to de-gas the liquid to eliminate all the air that might be trapped. During second cycle, class II HMX was added. This is the finer of the two classes of HMX and is added at this point to be coated evenly in the plastic binder. Half of the class III HMX was added during the third cycle. Its a coarser solid and this way, it helps to line up the particles properly to obtain a tighter packing density and therefore minimize voids. The last half of the class III HMX was added during the fourth cycle. The addition of the solid component was split up like this to get a good crystal distribution and thus would result in a higher packing density. What began as a liquid turned into a white, coarse, paste-like substance. The last cycle consisted of adding the isocyanate, which is a type of plastic that makes a polyurethane finish and securely binds the solid components together.

Each of the five batches were vacuum cast. These casts were then cured in an oven for two days to ensure that they were properly cured (solidified).

After the samples had cured, different manipulations were carried out, such as cutting, polishing, and tearing the samples for SEM analysis. Pictures were taken of the sample at all positions so that a collage could be made to represent the whole surface area. Five random lines were then drawn over the pictures and the crystals that were bisected by these lines were counted. These crystals were

then measured in microns and recorded under certain micron sizes according to the specifications. This way a number percentage was available and could be compared to the PBXN-110 specifications, to see if the mixes were consistent in particle representation.

Results

The particle analysis that was carried out on the five samples of Class III HMX showed that they did not match up exactly with the HMX specifications given within the MIL Standard. This probably was the result of not getting a representative sample from the bulk HMX containers. SEM was carried out on the HMX crystals. It clearly showed the crystal morphology and the undamaged HMX crystals.

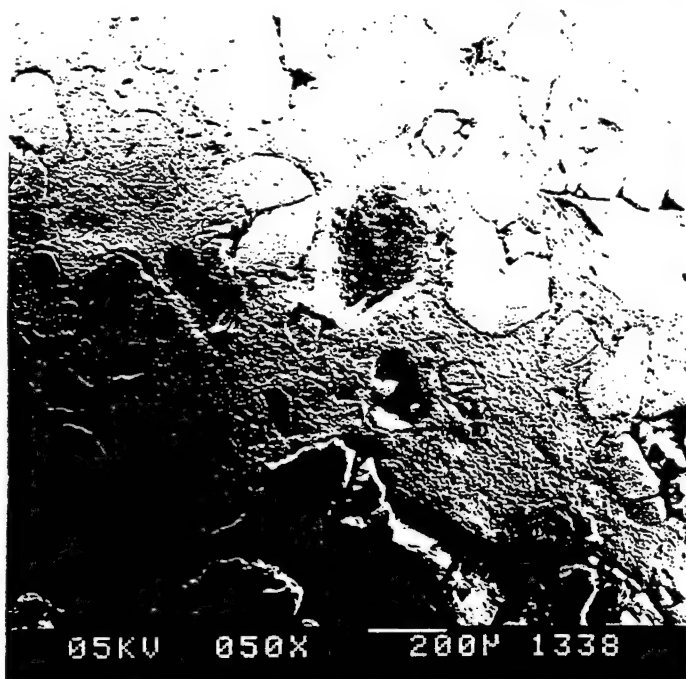
After the PBXN-110 had solidified, some samples were cut into cylinders and densities determined. Three different methods were used : Penta-Pycnometer, Mettler Density Determination apparatus, and hand calculations using length, diameter, and weight. The pycnometer uses helium gas displacement to determine the volume. The Mettler Density Determination apparatus uses water displacement. And finally, the hand calculations were carried out using the measured dimensions and the following formula: density is equal to mass/volume. After reviews of all the measurements it was determined that the water displacement method gave the most accurate readings. The average density value for the samples was $1.65 \pm 0.01 \text{ gm/cm}^3$.

Samples were then cut into appropriate dimensions and put into the SEM for analysis. Samples were cut, torn, or polished. After examining the SEM photographs, it was determined that polishing was the best technique for preparing the samples. The cutting caused extensive damage to the HMX crystals in the rubbery matrix. Tearing of the samples resulted in crystals being pulled away from the matrix, and thus disturbing the solid - matrix interaction. Polishing was determined to be the best technique for no significant damage was done to the crystals, and no apparent voids were introduced. The polishing technique needs refinement, e.g., a solvent used to dissolve away part of the crystals so that the smaller crystals will show up better.

These are pictures to show the Class III HMX crystals, the PBXN-110 polished samples, and the PBXN-110 torn samples. All these were taken by the SEM.



SEM Picture of HMX Class III Crystals



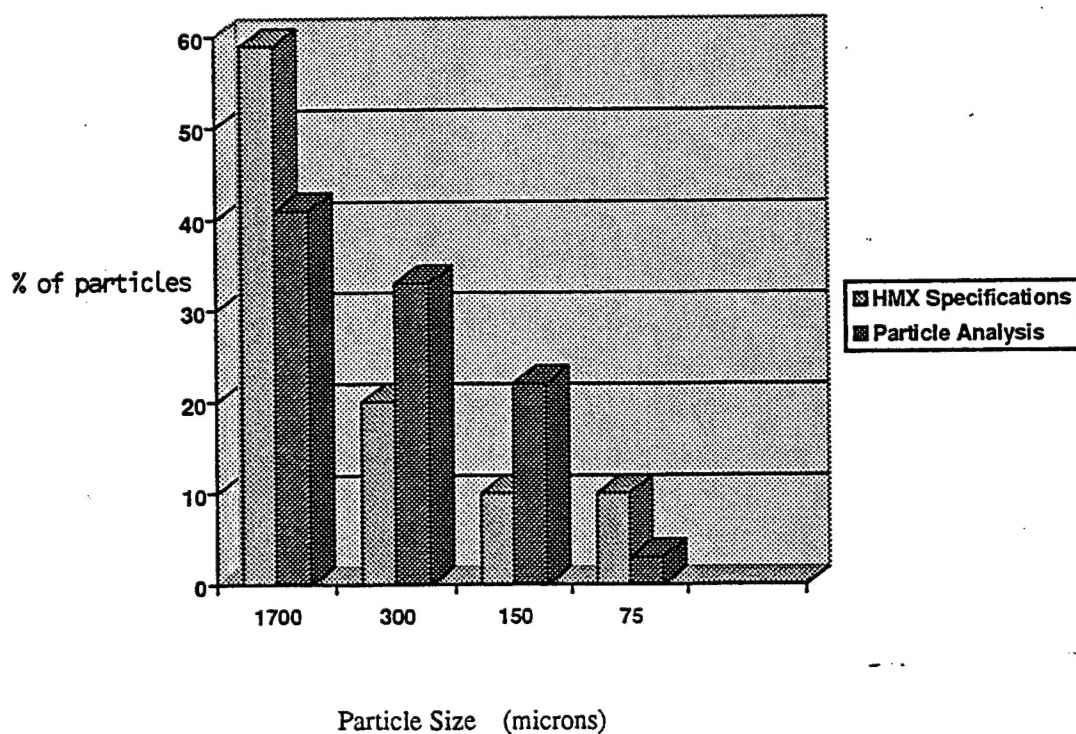
SEM Analysis of a Polished Sample of PBXN-110



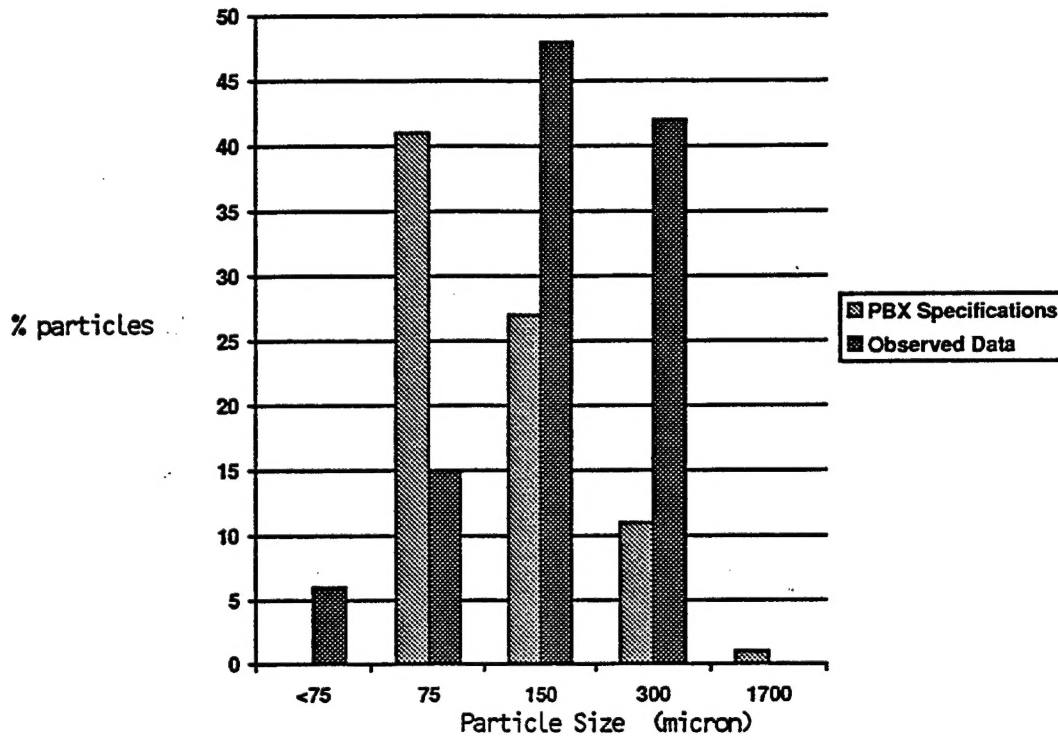
SEM Analysis of a Torn Sample of PBXN-110

After these pictures were taken and made into collages, physical measurements were carried out. On each picture, five random lines were drawn. Then each crystal that bisected these lines was recorded and measured. The result was a quantifiable number that can be compared to the PBXN-110 specifications. Graphs were made and compared and it showed that the torn samples deviated from the specifications more than the polished samples. This may be due to the shadows in the SEM pictures that resulted from the crystals that were torn out. The polished sample data matched the original particle distribution data.

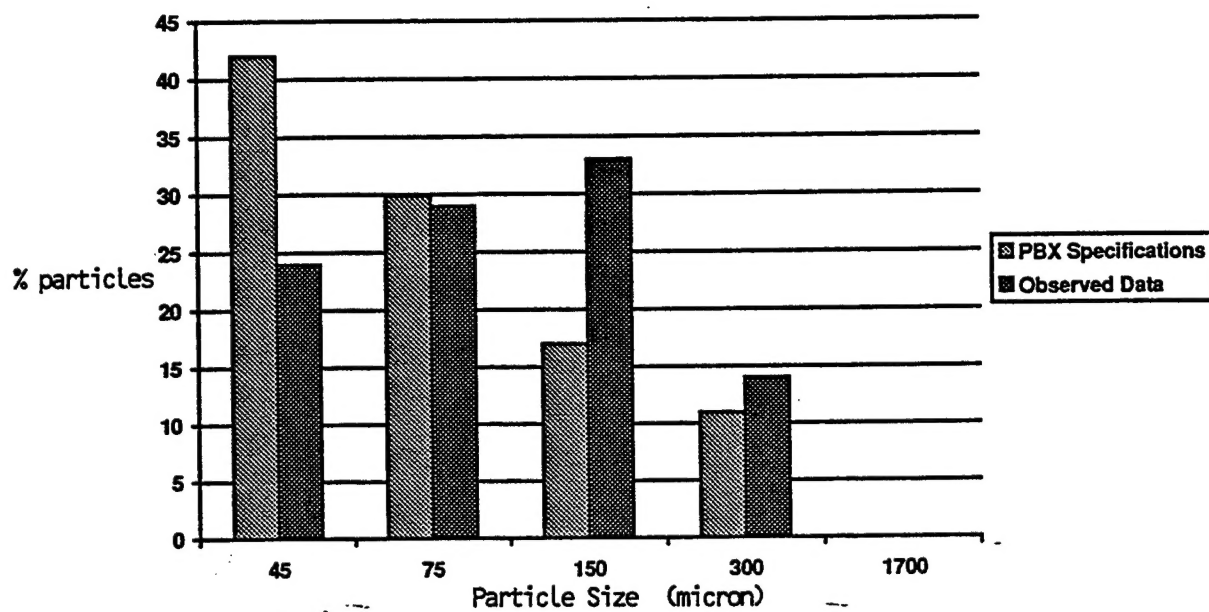
This graph below shows the comparison of our particle analysis to the HMX specifications. The numbers on the y-axis represent the percent out of one hundred. The numbers on the x-axis represent the particle size measured in microns.



This chart shows the comparison of the PBXN-110 specifications to the observed data , for a torn sample, from the physical measurements of the SEM photographs. The numbers on the y-axis represent the percent number and the numbers on the x-axis represent the particle size. This comparison is based on percent less then one hundred.



This comparison shows the PBXN-110 specifications to the observed data, for a polished sample, from the physical measurements of the SEM photographs. Again, the numbers on the y-axis represent the percent out of one hundred and the numbers on the x-axis represent the particle size in microns.



Conclusion

In conclusion, it was determined that quantifying the crystals in the overall matrix is possible. Cutting the samples causes an extensive amount of damage and polishing is the best technique to use. The polishing, though, could be more refined and gentler. It is also determined that the quality control is real important to the fabrication of small specimen.